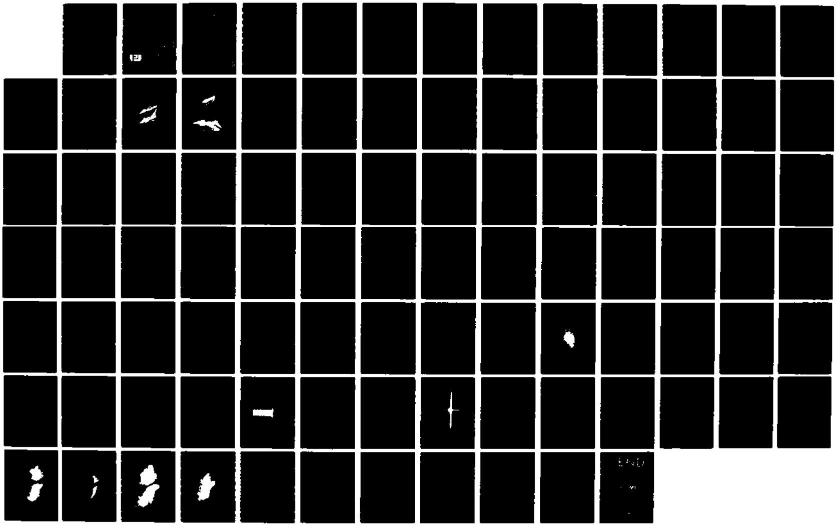
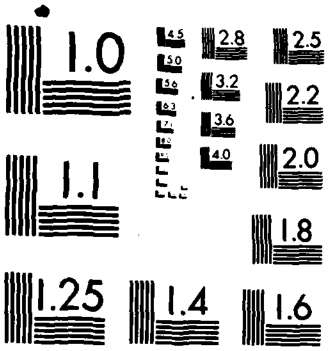
AD-A162 168 A ZONAL APPROACH TO THE DESIGN OF FINITE ELEMENT GRIDS 1/1
FOR 3-D TRANSONIC (U) INDIANA UNIV-PURDUE UNIV AT
INDIANAPOLIS SCHOOL OF ENGINEERIN A ECER 30 JUN 85
UNCLASSIFIED AFOSR-TR-85-1061 F49620-83-K-0034 F/G 20/4 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Annual
[redacted] Report

submitted to

Air Force Office
of Scientific Research
Bolling Air Force Base
Washington DC 20332

for

A Zonal Approach to the Design of Finite Element Grids for 3-D Transonic Flows with Complex Geometries

(Contract No: F49620-83-K-0034)

for the period of

June 1, 1983-June 30, 1985

PURDUE UNIVERSITY

School of Engineering
and Technology
at Indianapolis

Indiana University—Purdue University at Indianapolis

2

Annual
[redacted] Report

submitted to

Air Force Office
of Scientific Research
Bolling Air Force Base
Washington DC 20332

for

A Zonal Approach to the Design of Finite Element Grids for 3-D Transonic Flows with Complex Geometries

(Contract No: F49620-83-K-0034)

for the period of

June 1, 1983-June 30, 1985

by

Akin Ecer, Principal Investigator
Division of Engineering
Purdue University at Indianapolis
1201 E. 38th Street
Indianapolis, IN 46223

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release, distribution unlimited

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 85-1061

6a. NAME OF PERFORMING ORGANIZATION
PURDUE UNIVERSITY at INDIANAPOLIS

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
AFOSR/NA

6c. ADDRESS (City, State and ZIP Code)
SCHOOL OF ENGINEERING AND TECHNOLOGY
1202 E 38th STREET
INDIANAPOLIS, IN 46223

7b. ADDRESS (City, State and ZIP Code)
BOLLING AFB DC 20332-6448

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

8b. OFFICE SYMBOL (If applicable)
AFOSR/NA

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F49620-83-K-0034

8c. ADDRESS (City, State and ZIP Code)
BOLLING AFB, DC 20332

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2307	A4	

11. TITLE (Include Security Classification) A ZONAL APPROACH TO THE DESIGN OF FINITE ELEMENT GRIDS FOR 3D TRANSONIC FLOWS WITH COMPLEX GEOMETRIES

12. PERSONAL AUTHOR(S)
AKIN ECER

13a. TYPE OF REPORT
Annual

13b. TIME COVERED
FROM 1 Jun83 TO 31 May85

14. DATE OF REPORT (Yr., Mo., Day)
May 1985

15. PAGE COUNT
88

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FINITE ELEMENT METHOD, TRANSONIC FLOW, THREE DIMENSIONAL FLOW, POTENTIAL FLOW.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A block-structured solution scheme is developed for the analysis of three-dimensional transonic flows. The scheme is based on the solution of potential flow equations for individual blocks representing part of the flow field. Based on a previously developed block-structured grid generation scheme, appropriate computational grids are generated for each of the blocks depending on the complexity of the local flow field. The equations are then solved to provide a solution of a large problem in terms of an assembly of smaller problems for each block.

Numerical results illustrate the applicability of the method for a three-dimensional flow field around a wing profile (NACA0012). Different block structures are analyzed to demonstrate the robustness and the accuracy of the developed method. Finally a three-dimensional wing-body configuration is analyzed and the results are compared with previously obtained single block solutions.

The method is expandable to the solution of Euler and Navier-Stokes equations. It is also suited to be executed in a parallel processing environment. Keywords:

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
JAMES D WILSON

22b. TELEPHONE NUMBER (Include Area Code)
(202) 767-4935

22c. OFFICE SYMBOL
AFOSR/NA

TABLE OF CONTENTS

Page

ABSTRACT

A block-structured solution scheme is developed for the analysis of three-dimensional transonic flows. The scheme is based on the solution of potential flow equations for individual blocks representing part of the flow field. Based on a previously developed block-structured grid generation scheme, appropriate computational grids are generated for each of the blocks depending on the complexity of the local flow field. The equations are then solved to provide a solution of a large problem in terms of an assembly of smaller problems for each block.

Numerical results illustrate the applicability of the method for a three-dimensional flow field around a wing profile (NACA0012). Different block structures are analyzed to demonstrate the robustness and the accuracy of the developed method. Finally a three-dimensional wing-body configuration is analyzed and the results are compared with previously obtained single block solutions.

The method is expandable to the solution of Euler and Navier-Stokes equations. It is also suited to be executed in a parallel processing environment.

LIST OF FIGURES

I. INTRODUCTION

The solution of three-dimensional, transonic flows around complex aircraft configurations requires considerable computational effort. The capability to solve such large problems relies heavily on the availability of larger and faster computers. Over the last twenty years, considerable progress has been made in terms of hardware available for solving computational fluid dynamics problems. Rakich [1], in his introduction to the 1973 AIAA Computational Fluid Dynamics Conference, compared machines like IBM 360/67 to the kinds of CDC Star and Burroughs ILLIAC. At the time, parallel and vector machines were recently introduced and being applied to the solution of computational fluid dynamics problems. In 1983, further hardware development has resulted in vector machines like CRAY X MP or CYBER-205 [2]. Such comparisons may indicate the kind of progress in computer power we may expect in coming years. It should also be mentioned that these types of machines have also become much more widely available to the researchers in the field. In the next five years, we expect the major emphasis in hardware development to be in terms of multiprocessing capabilities and larger memories rather than drastic changes in computational speeds.

The computation of flow problems using vector machines has resulted in comparisons of computational speeds with other machines. For example, Chima and Johnson [3], compared Euler and Navier-Stokes solutions for transonic flows through a cascade of airfoils on two different computers: IBM 370/3033 and CRAY 1-S. In their comparisons, they have employed different levels of grid refinement for which they employed a Lax-Wendroff scheme. It can be seen from such comparisons, however, that the evaluation of the efficiency of a particular hardware configuration is not straightforward. It depends on the computational scheme and it is also strongly related to the computational grid employed in the analysis. It is possible to define a computational fluid dynamics problem on a box-type grid structure with MxNxK grid points in each direction. One can then specify the efficiency of a numerical scheme for such a rectangular grid structure and for a given hardware configuration.

A second problem to be considered is the number of grid points in a single block which can be fitted in a particular vector machine. If the strategy is to fit a single block with a regular grid structure around an aircraft geometry, the number of grid points required to model the complete flow field can grow very rapidly. Attempts are being made to improve this situation by embedding blocks with several levels of grid refinement inside a regular but coarser grid structure [4]. Until now, the most popular strategy for solving large computational fluid dynamics problems has been, a) to develop computational schemes which are fast on regular grids and, b) to model irregular geometries using regular grid structures.

Our work has been directed in applying finite element methods for solving fluid mechanics problems. This method provides an alternate approach over the ones discussed above for the solution of the same problem. One of the important features of this method is the possibility of using irregular grids that can be fitted efficiently to simulate irregular flow fields around complex geometries. It provides sufficient flexibility to design a computational grid around a complex aircraft geometry [5]. It also provides the freedom to modify an existing grid to provide a better approximation basis for the employed numerical scheme.

The applications of irregular grids, however, require development of numerical schemes where no regularity of the nodal connectivities is assumed. Our work has been aimed at developing such numerical schemes [6] where we can exploit the advantages of irregular yet more efficient grids. As it will be discussed in the following sections, rather than employing point or line relaxation schemes, we use block-relaxation schemes. Such schemes produce a uniform convergence rate for all of the eigenvalues of a differential operator inside a block. It is not affected by the occurence of an irregular grid structure inside a block. Such schemes have two basic constraints for solving large problems: a) the size of the block operator grows proportional to the size of the block and, b) the geometric definition of an irregular grid requires much more data than a regular grid, thus considerable information has to be generated or read-in during each relaxation step.

Based on the above considerations, we have been working on developing a block-by-block solution scheme. The main objective of this approach is to divide a large problem into smaller components in terms of a series of blocks. Rather than attempting to solve a large problem most efficiently on a single processor, we divide the problem into smaller ones and try to develop an "intelligent" strategy which is suitable for parallel processing. Parallel processing for large systems is a popular subject addressed by many researchers today. Our main objective is to exploit the physical characteristics of the problem where each block corresponds to a sub-volume in physical space. We developed a "sub-structuring" scheme where each of the "sub-structures" corresponds to a particular flow region. One can then design grids, use efficient solution schemes for each of the sub-regions depending on the characteristics of the flow field, and allocate computer resources in a parallel processing environment in a most efficient manner. We can summarize our progress in this area along the following lines:

- Development of a block-structured grid generation scheme [5],
- Development of a variational formulation for solving potential, Euler, and Navier-Stokes equations in a unified form [7], and
- Development of a block-structured solution scheme for solving the steady-state flow equations through a relaxation scheme.

In this report we will present only a short summary of the first two steps. We will mostly concentrate on the basic features of the block-structured solution scheme and its application to three-dimensional transonic potential flows.

A. Block-Structured Grid Generation Scheme

The generation of grids around a complex aircraft configuration is not a straight-forward task. It requires considerable understanding of the surrounding flow field. One has to provide a grid which will produce accurate results at all critical regions. Yet, excessive refinement of three-dimensional grids is not possible even when the largest computers are available. One has to be able to generate a computational grid in such a manner that a block of grid points can be controlled and modified after inspection. The optimum grid configuration for each of the critical flow regions can be quite different, yet, they have to be connected to each other. The grid generation techniques to be employed for this purpose should be sufficiently general to model any complex configuration.

In terms of generating a block-structured grid generation scheme, a finite element approach provides certain advantages. As discussed above, since irregular grids are allowed in a finite element solution scheme, it is easier to design grids individually for each block and then assemble them together. In comparison, the global mapping techniques used for grid generation requires additional computations to provide appropriate coupling of the blocks [8].

The developed block-structured grid generation scheme can be summarized in terms of the following steps: If the aircraft can be considered as as assembly of the body, engines, wings, and many other components, one has to provide appropriate grids for each of these components. Figure 1 illustrates the work performed for generating the details of the aircraft geometry. As can be seen in this figure, several sections are digitized to obtain the geometry and a three-dimensional representation of the body is obtained. A series of blocks are then constructed around the aircraft to model the flow field. The block structure for this problem is shown in Figure 2. As can be seen from this figure, the block structure is irregular, i.e., it includes openings between the blocks, it has irregular blocks (tetrahedrons) and finally some of the blocks are voids. The developed grid generation scheme can treat such an irregular block structure. One can design quite irregular grids in each of the blocks, yet attach the blocks in an organized fashion, as shown in Figure 3. Details of this approach can be found in reference [5].

The main objective of the work described in the present report is to determine the optimum computational procedure in terms of number of computations and number of data transfers for analyzing such irregular block structures. Generation of a block-structured grid provides the most natural way to develop a block-structured solution scheme. It also provides an insight for designing the most efficient solution scheme for a particular problem.

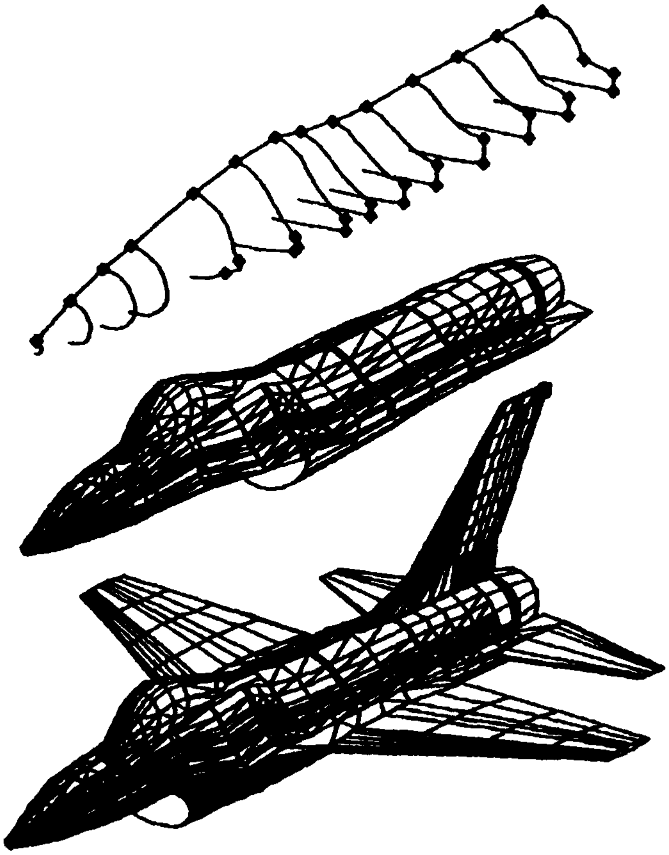


Figure 1. Construction of the aircraft geometry.

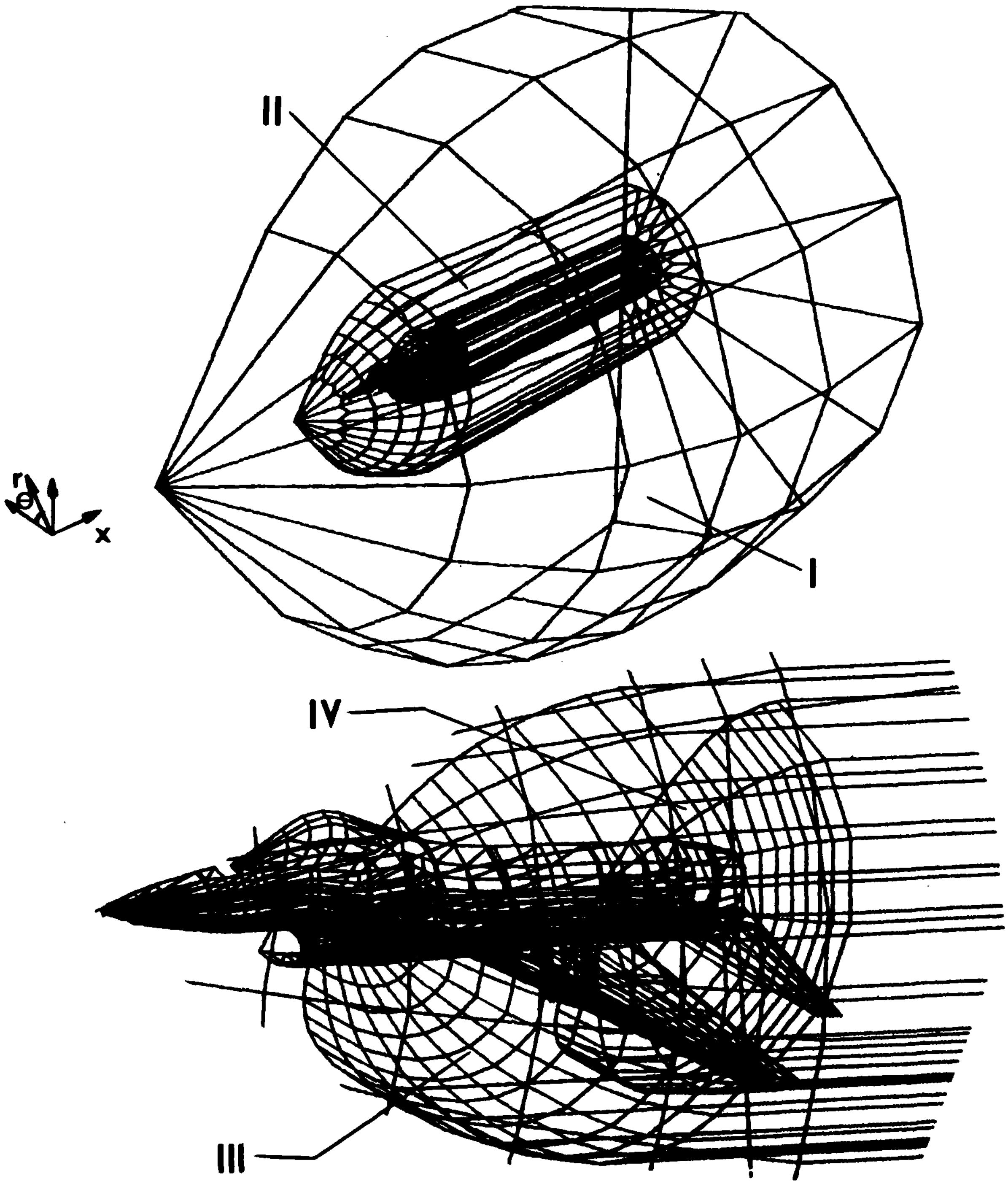


Figure 2. Construction of blocks around an aircraft geometry.

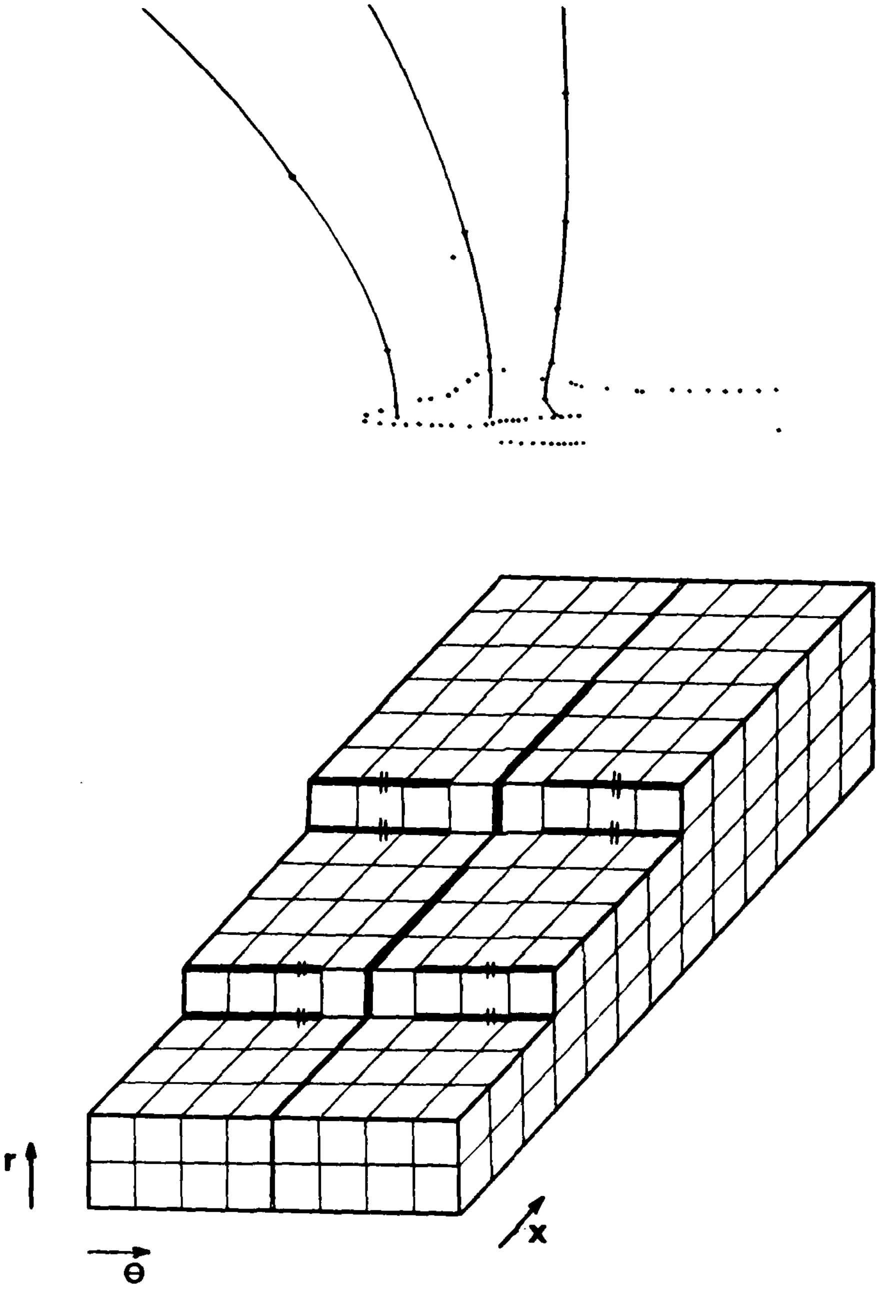
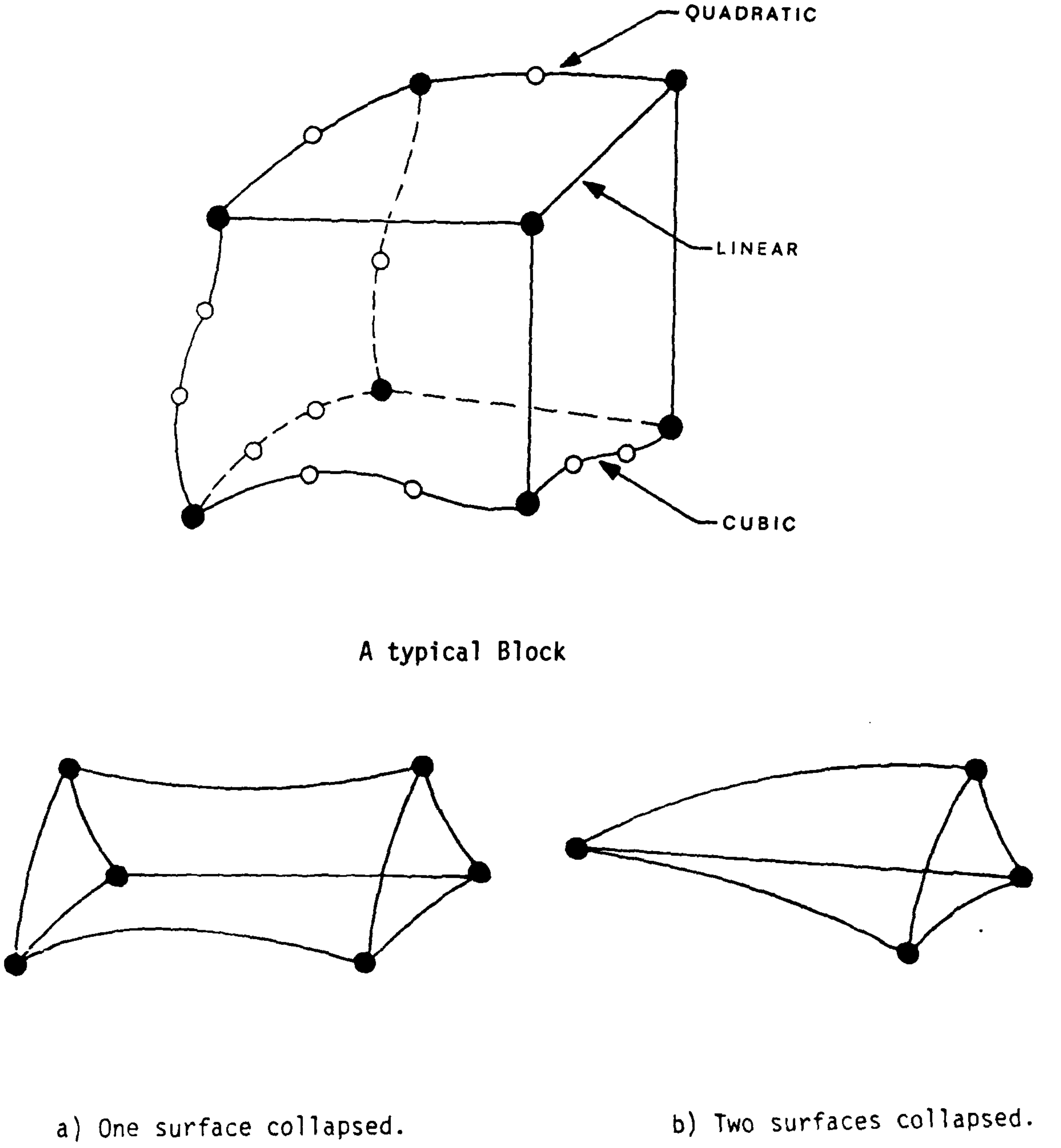


Figure 2. 'Continued.

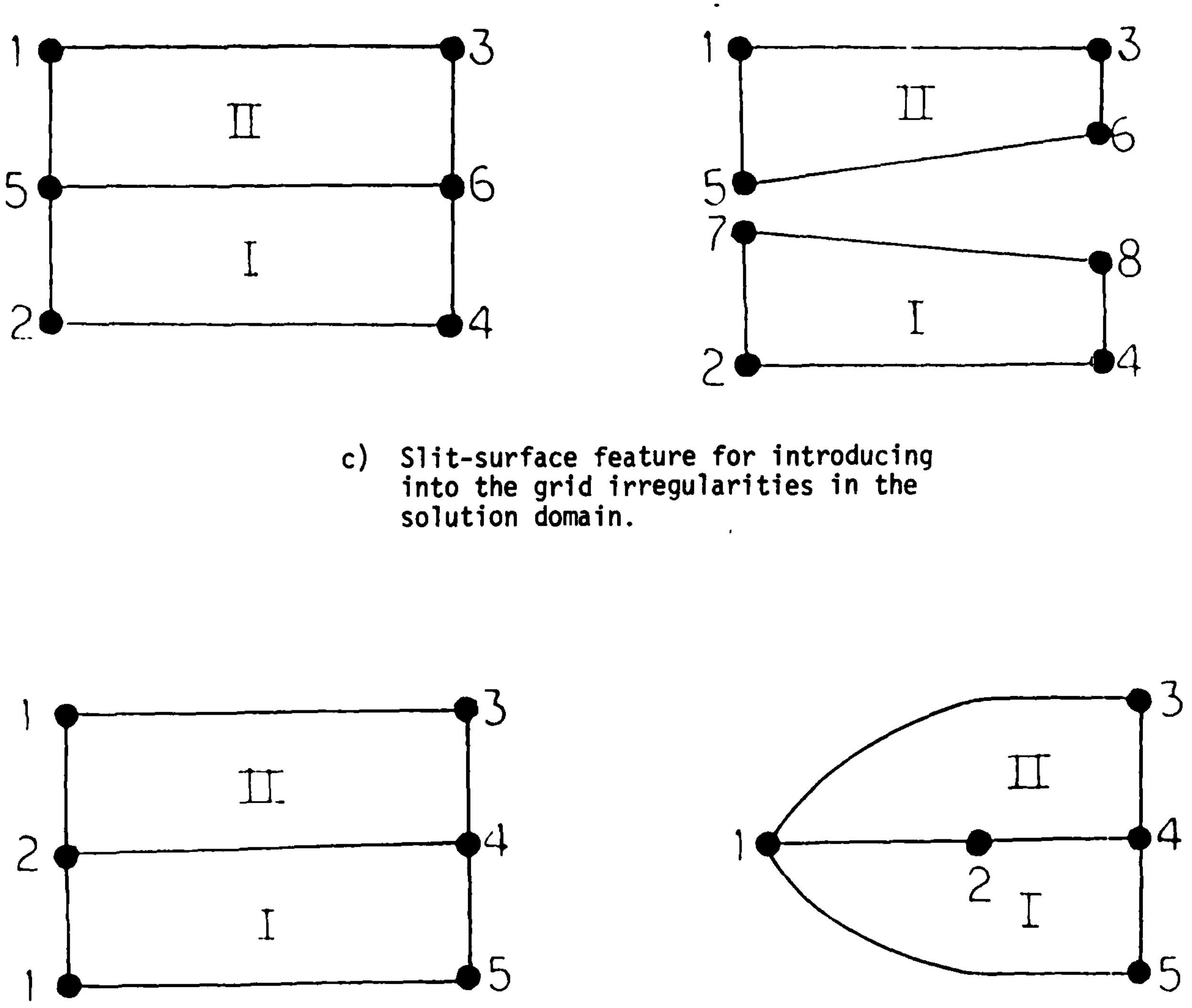


A typical Block

a) One surface collapsed.

b) Two surfaces collapsed.

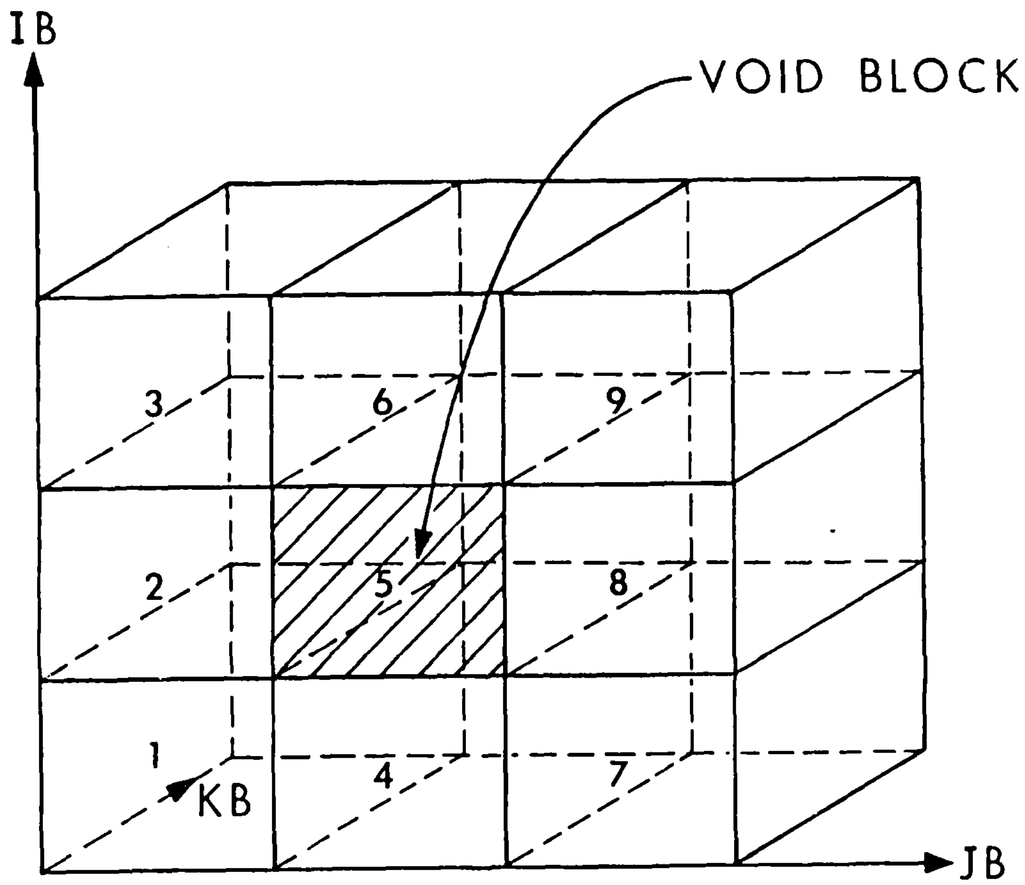
Figure 3. Capabilities of grid generation scheme for modeling irregular block structures.



c) Slit-surface feature for introducing into the grid irregularities in the solution domain.

d) Coupled-surface feature for treating radial-type grids.

Figure 3. Continued.



e) The 'void block' feature.

Figure 3. 'Continued.

B. Solution of Flow Problems in Terms of Clebsch Variables

The ultimate objective in computational fluid mechanics is to solve Navier-Stokes equations for three-dimensional turbulent flows. Since such a proposal is computationally costly, a general and yet quite efficient approach is to solve these equations only in flow regions where viscous effects are important. One of the most successful applications of computational fluid mechanics has been in the area of solving potential flow problems together with boundary layer corrections. We have attempted to achieve such a simplification in a general manner by using a new formulation of the physical problem. Rather than using the velocities and pressure as the primitive variables, we decomposed the velocity vector into irrotational, rotational-inviscid and viscous components by using a Clebsch transformation [9]. This enables the solution of the general problem in terms of potential, Euler, and Navier-Stokes approximations at different flow regions which in our case are defined as blocks. The objective is to provide general yet efficient solution schemes as well as appropriate computational grids in a block-structured flow domain for all three cases.

For analyzing potential flows, it is common to employ the velocity potential as the primitive variable. One can write the velocity vector in terms of a velocity potential as follows:

$$\underline{u} = \underline{\nabla}\phi \tag{1}$$

and proceed to solve the conservation of mass equation. In this case, the conservation of momentum and energy are automatically satisfied [7]. A second step in the approximate solution of flow problems is the Euler equations. In this case, one can write the velocity field in terms of the entropy and total enthalpy as follows:

$$\underline{u} = \underline{\nabla}\phi + H\underline{\nabla}\eta + S\underline{\nabla}\lambda \tag{2}$$

where ϕ, η, λ are a set of Lagrange multipliers [9]. Here the equations for the conservation of mass, entropy and total enthalpy have to be solved. Finally, for viscous flows, the velocity field can be written in terms of:

$$\underline{u} = \underline{\nabla}\phi + H\underline{\nabla}\eta + S\underline{\nabla}\lambda + \text{viscous terms.} \tag{3}$$

Here the viscous dissipation has to be calculated together with the shear stresses on the solid walls for a given velocity field.

In the foregoing three levels of approximation, one can observe that the number of variables and the number of equations increases parallel to the complexity of the model. If we assume that in one flow region (a block), we will be solving Euler equations,while at a neighboring block the potential flow equations are sufficiently accurate, the interface boundary requires specification of S and H to be a constant along the boundary. Only one equation is solved in the irrotational flow region, while three equations are solved in the block with the rotational flow.

Another point to be remembered is the relationship between the flow models and the computational grids. One can easily realize that more complicated models require much finer grids. For example, the solution of two-dimensional, transonic potential flows around a semi-circle requires a much coarser grid than the one for solving the Euler equations for the same geometry [9]. If the convection of vorticities generated by the shock is considered, one has to design much finer grids to capture such a phenomenon.

C. The Necessity for Block-Structured Solution Schemes

For the development of a block-structured scheme the methodology described above was employed. The computational strategy is based on the capability of generating grids with varying degrees of refinement for each block and solve potential, Euler, or Navier-Stokes equations for each block depending on the characteristics of the flow field. As it will be discussed later, accurate and efficient implementation of the boundary conditions between neighboring blocks is extremely important in a block-structured scheme.

The most important consideration which necessitates the use of a block-structured solution scheme is the size of the problem. We do not expect to have computers large enough to fit the problems we are interested in into their main memory during the next five years. In terms of storing the geometry information describing an irregular geometry in a general form, data transfer becomes a major problem. If one employs auxiliary storage to store large amounts of data, the computational scheme can easily become I/O bound when large amounts of data have to be transferred. Depending on the size of the problem and the type of information to be stored, one can even sometimes deplete these types of storage facilities.

The approach we have taken here is to develop a block-structured solution scheme, where it is possible to use the available computer resources in a most efficient manner. For a given problem one has to make decisions in terms of designing a computational grid. Depending on the size of the problem, the computer resources available, and the characteristics of the flow field, we can decide on the size of the blocks. In the developed numerical procedure, based on physical insight, one can decide the manner in which iterations should be performed and the data should be transferred between the blocks. If we have a supersonic pocket developing inside a single block, we may choose a particular iterative procedure for solving this problem. We may decide to iterate on that block more often than the others. The objective here is to develop "intelligent" schemes where efficiency can be improved as we learn more about the details of the fluid mechanics problem and we can plan and revise our computational strategy.

In this report we will discuss only the solution of the potential equation using a block-structured solution scheme. However, we will comment on its generality and its applications to more complex flow models.

II. A BLOCK-STRUCTURED SOLUTION SCHEME FOR POTENTIAL FLOWS

A. Potential Flow Problem

The analysis of compressible, irrotational flows requires only the solution of the following conservation of mass equation:

$$\underline{\nabla}(\rho \underline{u}) = 0 \quad \text{in } \Omega \text{ ,} \tag{4}$$

where $\underline{u}$ is the velocity vector, ρ is the density and Ω is the flow region to be analyzed. The density is a function of the local velocity which can be written as follows:

$$\rho = C(K^2 - \underline{u} \cdot \underline{u})^{\frac{1}{\gamma - 1}} \tag{5}$$

In the above equation, C, K and γ are known constants. The boundary conditions require the specification of the normal mass flux on the boundary surface Γ.

$$\rho \underline{u} \cdot \underline{n} = f \qquad \text{on } \Gamma \tag{6}$$

where f is a known function specified on the boundary and $\underline{n}$ is the unit normal vector. By using the condition for irrotationality, one can substitute the velocity potential to eliminate the velocity vector as follows:

$$\underline{u} = \underline{\nabla} \Phi \tag{7}$$

The conservation of mass equation then becomes second-order,

$$\underline{\nabla} \cdot (\rho \underline{\nabla} \Phi) = 0 \qquad \text{in } \Omega \tag{8}$$

where the velocity potential has to be assigned an arbitrary value at one point to remove the singularity due to the introduction of equation (7). Since, the conservation of mass equation has to be satisfied for the entire flow domain, over a closed boundary, i.e.,

$$\oint_\Gamma \rho \underline{u} \cdot \underline{n} \, d\Gamma = 0 \tag{9}$$

the specified value of Φ can be arbitrary.

The objective of the present analysis is to develop a relaxation scheme for obtaining a solution to the above steady-flow equations.

B. Formulation of the Problem in a Block-Structured Form

For developing a block-structured solution scheme, we will define the problem over a series of flow regions which we will call blocks in the following form:

$$\underline{\nabla}\cdot(\rho\underline{\nabla}\Phi_i) = 0 \qquad \text{in } \Omega_i$$

$$\rho\underline{n}\cdot\underline{\nabla}\Phi_i = f_i \qquad \text{on } \Gamma_i \tag{10}$$

when Ω_i and Γ_i are the block volume and block surface for each block, where i indicates the block number. The boundaries can be further classified in two parts: the boundaries between the neighboring blocks and the global boundary of the entire flow region. We can distinguish the corresponding boundary conditions as follows:

$$\rho\underline{n}\cdot\underline{\nabla}\Phi_i = f_i \qquad \text{on } \Gamma_i^{\,o} \tag{11}$$

$$\rho\underline{n}\cdot\underline{\nabla}\Phi_i = g_i \qquad \text{on } \Gamma_i^{\,c} \tag{12}$$

where $\Gamma_i^{\,o}$ is the global boundary and $\Gamma_i^{\,c}$ is the inter-block boundary for the ith block. Of course, some blocks may have no connection to global boundaries.

The problem can then be defined as the solution of conservation of mass equation (10) for each block, together with the determination of unknown inter-block boundary fluxes g_i. This can be achieved through an additional constraint which specifies that the velocity potentials are continuous across the inter-block boundaries. We can define the corresponding variational problem in the following form:

$$\delta\pi = \sum_i \int_{V_i} \rho\underline{\nabla}\Phi_i\cdot\underline{\nabla}\delta\Phi_i \, dV + \int_{\Gamma_i^o} f_i\delta\Phi_i \, d\Gamma + \int_{\Gamma_i^c} g_i\delta\Phi_i \, d\Gamma = 0 \tag{13}$$

By taking variations with respect to $\underline{\Phi}$, we can derive the differential equation in equation (9) and the boundary conditions in equations (11) and (12). Also, since we assumed that g_i is an unknown, we can take a variation with respect to g_i. This produces the additional constraint equation in terms of the compatibility of velocity potentials across neighboring block boundaries as follows:

$$\sum_i \int_{\Gamma_i^c} \underline{\Phi}_i \; \delta g_i \; d\Gamma = 0 \qquad (14)$$

Since, the same g_i is employed for neighboring blocks, this constraint will involve the velocity potentials which correspond to the grid points on such blade pairs as follows:

$$\underline{\Phi}_{i,j} - \underline{\Phi}_{n,j} = 0 \qquad (15)$$

where i and n are the two neighboring blocks and j is the number for the boundary surface between these two blocks.

At this point one can introduce the finite element formulation, where the flow region is divided into first a series of blocks and then a group of finite elements. The distribution of velocity potential over each element for block number (i) is approximated by:

$$\Phi_i(x,y,z) = N_k(x,y,z) \; \Phi_{ik} \qquad (16)$$

where N_k is called a shape function (a simple polynomial) and Φ_k is the nodal value of the velocity potential at node number k. One can also approximate the unknown boundary fluxes along the block interfaces in terms of nodal flux values as follows:

$$g_j(s,t) = \tilde{N}_k(s,t) \; \lambda_{jk} \qquad (17)$$

where $\tilde{N}_k$ is a two-dimensional shape function for node k and (s,t) are the two-dimensional coordinates along the boundary surfaces. After substituting the finite element approximations and taking variation with respect to nodal values of velocity potentials, one can write the following set of discrete equations

$$\underline{A}_i \, \underline{\Phi}_i + \underline{C}^t_{i,j} \, \underline{\lambda}_j = \underline{F}_i \qquad (18)$$

Where $\underline{A}$, $\underline{F}$, $\underline{C}$, $\underline{\Phi}$ and $\underline{\lambda}$ are global matrices for each block (i) defined by assembling element matrices for each element e.

$$\underline{A}_{i,e} = \int_{V_{i,e}} \rho(\underline{N},_x \underline{N},_x^t + \underline{N},_y N,_y^t + \underline{N},_z \underline{N},_z^t) \, dV \qquad (19)$$

$$\underline{F}_{i,e} = \int_{\Gamma^o_{i,e}} \rho_e f_e \, \underline{N} \, d\Gamma \qquad (20)$$

$$\underline{C}_{i,j,e} = - \int_{\Gamma^c_{i,e}} \rho_e \tilde{\underline{N}} \, \underline{N}^t \, d\Gamma \qquad (21)$$

$$\underline{\lambda}_j = [\lambda_k]_j \qquad (22)$$

Here, i indicates the number of the block, j indicates the number of the sui face, k is the node number and e is the element number. For $\rho > o$, the coefficient matrix in equation (19) is symmetric and positive-definite.

The variation with respect to nodal values of boundary fluxes (λ_k), produces a set of constraint equations defining the compatibility of velocity potentials between the blocks:

$$\sum_i \underline{C}_{i,j} \, \underline{\Phi}_i = 0 \qquad (23)$$

As can be seen from the above equation, matrix $\underline{C}_{i,j}$ picks up all the nodes on surface j located on the block i and assembles the constraint conditions. In coupled form equations (17) and (20) can be written as follows:

$$\begin{bmatrix} [\underline{A}_i] & [\underline{C}^t_{i,j}] \\ [\underline{C}_{i,j}] & [\underline{0}] \end{bmatrix} \begin{bmatrix} [\underline{\Phi}_i] \\ [\underline{\lambda}_j] \end{bmatrix} = \begin{bmatrix} [\underline{F}_i] \\ [\underline{0}] \end{bmatrix} \qquad (24)$$

For a block structure shown in Figure 4, the equation (24) can be written in the following form:

$$\begin{bmatrix} \underline{A}_1 & \underline{0} & \underline{0} & \underline{C}^t_{1,1} & \underline{0} \\ \underline{0} & \underline{A}_2 & \underline{0} & \underline{C}^t_{2,1} & \underline{C}^t_{2,2} \\ \underline{0} & \underline{0} & \underline{A}_3 & \underline{0} & \underline{C}^t_{3,2} \\ \underline{C}_{1,1} & \underline{C}_{2,1} & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{C}_{2,2} & \underline{C}_{3,2} & \underline{0} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\Phi}_1 \\ \underline{\Phi}_2 \\ \underline{\Phi}_3 \\ \underline{\lambda}_1 \\ \underline{\lambda}_2 \end{bmatrix} = \begin{bmatrix} \underline{F}_1 \\ \underline{F}_2 \\ \underline{F}_2 \\ \underline{0} \\ \underline{0} \end{bmatrix} \qquad (25)$$

Here, there are three-blocks and two inter-connecting surfaces. In the above equation, the coupling matrices for each row of λ equations represent the contribution to a surface from the nodes located on the two neighboring blocks. As indicated above, unknowns are the nodal velocity potentials Φ_k and the nodal boundary fluxes λ_k. ($\underline{\Phi}_i$, $i = 1,3$) are the potential vectors for each block and ($\underline{\lambda}_j$, $j = 1,2$) are the vectors for each surface.

In general, the surfaces can be described by an independent set of nodes and the nodal values of $\underline{\lambda}_j$ vectors do not have to be necessarily positioned on the same physical point in space as of the elements of the $\underline{\Phi}$ vectors. However, in the applications presented in this paper, the same grid point locations were used for both $\underline{\Phi}$ and $\underline{\lambda}$ vectors.

C. Basic Considerations for the Development of an Iterative Solution Scheme for Block-Structured Equations

One can, of course, attempt to solve the coupled equations directly in the form of equation (24). Matrices $\underline{A}_i$ correspond to individual coefficient matrices for each block. Most of the computational effort in solving such a system will involve the treatment of the coupling matrices

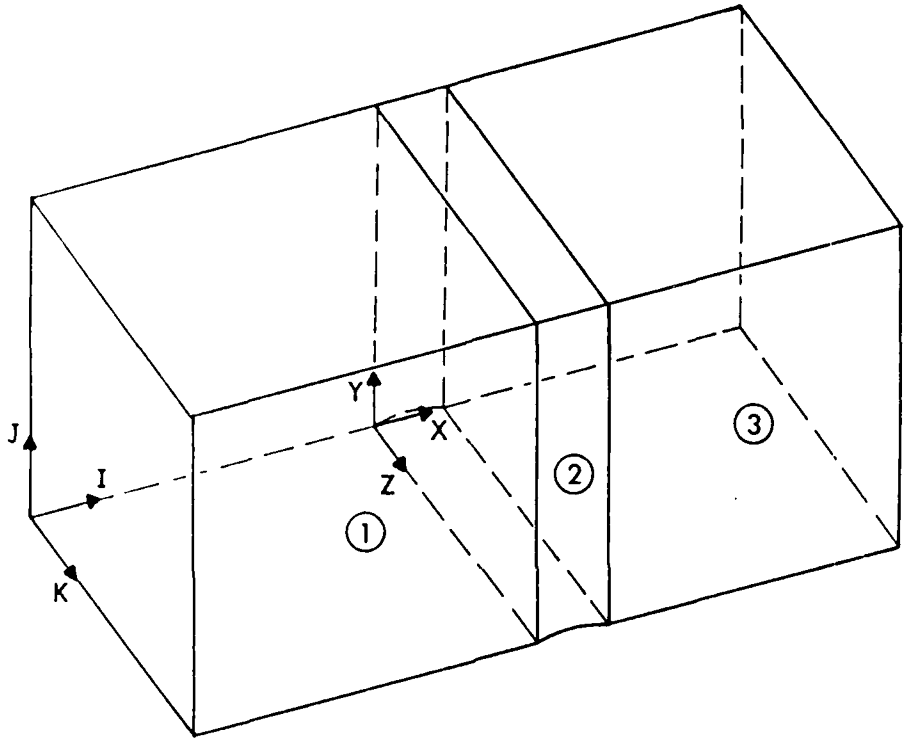


Figure 4. Three-block structure for a sample problem.

between the blocks. In the developed procedure, we attempt to solve the coupled equations by using an iterative procedure. For the description of this iterative procedure, let us consider a simpler system consisting of two blocks and a single coupling surface. In this case, the equations can be written as follows:

$$\begin{bmatrix} \underline{A}_1 & \underline{0} & \underline{C}^t_{1,1} \\ \underline{0} & \underline{A}_2 & \underline{C}^t_{2,1} \\ \underline{C}_{1,1} & \underline{C}_{2,1} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\Phi}_1 \\ \underline{\Phi}_2 \\ \underline{\lambda}_1 \end{bmatrix} = \begin{bmatrix} \underline{F}_1 \\ \underline{F}_2 \\ \underline{0} \end{bmatrix} \tag{26}$$

Here $\underline{A}_1$, and $\underline{A}_2$ represent the Laplace operator for the two blocks. $\underline{C}_{1,1}$ and $\underline{C}_{2,1}$ are matrices for defining the coupling between the two blocks. Let us also consider a linearized discrete variational problem in the following form:

$$\tilde{\Pi} = \frac{1}{2}\, \underline{\Phi}_1^t \underline{A}_1 \underline{\Phi}_1 + \frac{1}{2}\, \underline{\Phi}_2^t \underline{A}_2 \underline{\Phi}_2 + \underline{\lambda}_1^t (\underline{C}_{1,1}\, \underline{\Phi}_1 + \underline{C}_{2,1}\, \underline{\Phi}_2)$$

$$- \underline{\Phi}_1^t \underline{F}_1 - \underline{\Phi}_2^t \underline{F}_2 \tag{27}$$

It can be shown that the variation of the above functional with respect to $\underline{\Phi}_1$, $\underline{\Phi}_2$ and λ produces the above set of equations.

At this point we can define a second variational problem as follows: Instead of the unknown flux vector $\underline{\lambda}$, we assume a known flux vector $\underline{\lambda}^*$. Then we can write the following variational problem:

$$\tilde{\Pi}^* = \frac{1}{2}\, \underline{\Phi}_1^{*t} \underline{A}_1 \underline{\Phi}_1^* + \frac{1}{2}\, \underline{\Phi}_2^{*t} \underline{A}_2 \underline{\Phi}_2^* + \underline{\lambda}_1^{*t} (\underline{C}_{1,1} \underline{\Phi}_2^* + \underline{C}_{2,1} \underline{\Phi}_2^*)$$

$$- \underline{\Phi}_1^{*t} \underline{F}_1 - \Phi_2^{*t} \underline{F}_2 \tag{28}$$

Taking variation with respect to $\underline{\Phi}_1^*$ and $\underline{\Phi}_2^*$, we can write:

$$\underline{A}_1\underline{\Phi}_1^* + \underline{C}_{1,1}^t\lambda_1^* = \underline{F}_1$$

$$\underline{A}_2\underline{\Phi}_2^* + \underline{C}_{2,1}^t\underline{\lambda}_1^* = \underline{F}_2 \qquad (29)$$

We can prove a certain Theorem, from the solution of the above two variational problems:

Theorem. If $\underline{\lambda}^*$ is any arbitrary flux distribution on a boundary surface, the solution of equation (29) produces upper and lower bounds to the solution of the original problem of equation (26).

The validity of the above statement can now be shown. The solution of the above equations can be written as:

$$\underline{\Phi}_1^* = \underline{A}_1^{-1}\underline{F}_1 - \underline{A}_1^{-1}\underline{C}_{1,1}^t\underline{\lambda}_1^*$$

$$\underline{\Phi}_2^* = \underline{A}_2^{-1}\underline{F}_2 - \underline{A}_2^{-1}\underline{C}_{2,1}^t\underline{\lambda}_1^* \qquad (30)$$

The nodal potential vectors which correspond to the nodes on the interface can be written from the above solutions as follows:

$$\underline{C}_{1,1}\underline{\Phi}_1^* = \underline{C}_{1,1}\underline{A}_1^{-1}\underline{F}_1 - \underline{C}_{1,1}\underline{A}_1^{-1}\underline{C}_{1,1}^t\underline{\lambda}^*$$

$$\underline{C}_{2,1}\underline{\Phi}_2^* = \underline{C}_{2,1}\underline{A}_2^{-1}\underline{F}_2 - \underline{C}_{2,1}\underline{A}_2^{-1}\underline{C}_{2,1}^t\underline{\lambda}^* \qquad (31)$$

We can write the exact solution of equation (26) in a similar form,

$$\underline{C}_{1,1}\underline{\Phi}_1 = \underline{C}_{1,1}\underline{A}_1^{-1}\underline{F}_1 - \underline{C}_{1,1}\underline{A}_1^{-1}\underline{C}_{1,1}^t\underline{\lambda}_1$$

$$\underline{C}_{2,1}\underline{\Phi}_1 = \underline{C}_{2,1}\underline{A}_2^{-1}\underline{F}_2 - \underline{C}_{2,1}\underline{A}_2^{-1}\underline{C}_{2,1}^t\underline{\lambda}_1 \qquad (32)$$

The difference of equations (31) and (32) produce the following relationship:

$$\underline{C}_{1,1}(\underline{\Phi}_1^* - \underline{\Phi}_1) = \underline{C}_{1,1}\underline{A}_1^{-1}\underline{C}_{1,1}^t(\underline{\lambda} - \underline{\lambda}^*)$$

$$\underline{C}_{2,1}(\underline{\Phi}_2^* - \underline{\Phi}_2) = \underline{C}_{2,1}\underline{A}_2^{-1}\underline{C}_{2,1}^t(\underline{\lambda} - \underline{\lambda}^*) \qquad (33)$$

If we assume that $\underline{\Phi}_1^*$ and $\underline{\Phi}_2^*$ are close to the exact solution and can be written in the following form:

$$\underline{\Phi}_1^* = k_1\underline{\Phi}_1 + \text{small terms}$$

$$\underline{\Phi}_2^* = k_2\underline{\Phi}_2 + \text{small terms} \qquad (34)$$

where k_1 and k_2 are unknown constants, equation (33) can be re-written as follows:

$$(k_1 - 1)\ \underline{C}_{1,1}\ \underline{\Phi}_1 = \underline{C}_{1,1}\underline{A}_1^{-1}\underline{C}_{1,1}^t\ (\underline{\lambda} - \underline{\lambda}^*)$$

$$(k_2 - 1)\ \underline{C}_{2,1}\ \underline{\Phi}_2 = \underline{C}_{2,1}\underline{A}_2^{-1}\underline{C}_{2,1}^t\ (\underline{\lambda} - \underline{\lambda}^*) \qquad (35)$$

By using the compatability condition for the exact solution from equation (26) as follows:

$$\underline{C}_{1,1}\ \underline{\Phi}_1 + \underline{C}_{2,1}\ \underline{\Phi}_2 = 0 \quad , \qquad (36)$$

and since both matrices $\underline{A}_1$ and $\underline{A}_2$ are positive definite and by using equation (35) we can show that when

$$k_1 > 1 \quad \rightarrow \quad k_2 < 1 \qquad (37)$$

or

$$k_1 < 1 \quad \rightarrow \quad k_2 > 1 \quad . \qquad (37)$$

i.e., $\underline{\Phi}_1^*$ and $\underline{\Phi}_2^*$ provide upper and lower bounds to the solution of velocity potentials on the boundary.

A similar proof can be made for a second problem. This time, let us assume that we start with a set of approximate velocity potential vectors for each block, $\underline{\Phi}_1^*$ and $\underline{\Phi}_2^*$, such that the compatability condition at the boundaries is satisfied as follows:

$$\underline{C}_{1,1}\ \underline{\Phi}_1^* + \underline{C}_{2,1}\ \underline{\Phi}_2^* = 0 \qquad (38)$$

which means that for the approximate solution, the nodes at both sides of the interface between two blocks possess the same $\underline{\Phi}$ distribution, yet equation (32) is not satisfied. In this case, by solving equation (38) together with the following set of equations,

$$\underline{A}_1\underline{\Phi}_1^* + \underline{C}_{1,1}^t\underline{\lambda}_{1,1}^* = \underline{F}_1$$

$$\underline{A}_2\underline{\Phi}_2^* + \underline{C}_{2,1}^t\underline{\lambda}_{1,2}^* = \underline{F}_2 \qquad (39)$$

we obtain a different set of flux vectors $\underline{\lambda}_{1,1}^*$ and $\underline{\lambda}_{1,2}^*$ on each side of the boundary surface. If we solve equation (39) for the velocity potential vector as

$$\underline{\Phi}_1^* = \underline{A}_1^{-1}\underline{F}_1 - \underline{A}_1^{-1}\underline{C}_{1,1}^t\underline{\lambda}_{1,1}^*$$

$$\underline{\Phi}_2^* = \underline{A}_2^{-1}\underline{F}_2 - \underline{A}_2^{-1}\underline{C}_{2,1}^t\underline{\lambda}_{1,2}^* \qquad (40)$$

and substitute into equation (38), we can write the following relationship:

$$\underline{C}_{1,1}\underline{A}_1^{-1}\underline{F}_1 - \underline{C}_{1,1}\underline{A}_1^{-1}\underline{C}_{1,1}^t\underline{\lambda}_{1,1}^* + \underline{C}_{2,1}\underline{A}_2^{-1}\underline{F}_2 - \underline{C}_{2,1}\underline{A}_2^{-1}\underline{C}_{2,1}^t\underline{\lambda}_{1,2}^* = 0 \qquad (41)$$

On the other hand, the summation of the two equations in equation (32) produces the following relationship

$$\underline{C}_{1,1}\underline{\Phi}_1 + \underline{C}_{2,1}\underline{\Phi}_2 = \underline{C}_{1,1}\underline{A}_1^{-1}\underline{F}_1 - \underline{C}_{1,1}\underline{A}_1^{-1}\underline{C}_{1,1}^t\underline{\lambda}_1 + \underline{C}_{2,1}\underline{A}_2^{-1}\underline{F}_2$$

$$- \underline{C}_{2,1}\underline{A}_2^{-1}\underline{C}_{2,1}^t\underline{\lambda}_{1,1} \qquad (42)$$

Combining equations (41) and (42) one can then write,

$$\underline{C}_{1,1}\underline{A}_1^{-1}\underline{C}_{1,1}^t\,(\underline{\lambda}_1 - \underline{\lambda}_{1,1}^*) = - \underline{C}_{2,1}\underline{A}_2^{-1}\underline{C}_{2,1}^t\,(\underline{\lambda}_1 - \underline{\lambda}_{1,2}^*) \qquad (43)$$

Again by making a similar approximation as in the case of equation (35) and by recognizing that both matrices $\underline{A}_1$ and $\underline{A}_2$ are positive-definite, one can show that vectors $\underline{\lambda}_{1,1}^*$ and $\underline{\lambda}_{1,2}^*$ produce upper and lower bounds to the correct flux vector at the boundaries which can be stated as follows:

<u>Theorem</u>. If $\underline{\Phi}^*$ is any arbitrary velocity potential distribution on a boundary surface, the solution of equation (39) produces upper and lower bounds to the solution of the original problem of equation (26).

Based on the above discussions, an iteration procedure was developed using the following principles:

- calculate $\underline{\Phi}_1^*$ and $\underline{\Phi}_2^*$ from equation (29) which provides bounds for the solution at the boundary ($\underline{C}_1\underline{\Phi}$ and $\underline{C}_1\underline{\Phi}$),
- calculate an estimate of $\underline{\Phi}$ at the boundaries based on $\underline{\Phi}_1^*$ and $\underline{\Phi}_2^*$,
- solve equation (25) by assuming the velocity potentials at the boundaries. Since $\underline{\Phi}$ values at the boundaries are specified, $\underline{\lambda}$ values are not necessary at this step,
- using equations (38) and (39), calculate $\underline{\lambda}_{1,1}^*$ and $\underline{\lambda}_{2,1}^*$ which provide bounds for the next estimate of $\underline{\lambda}$,
- calculate a new estimate of $\underline{\lambda}^*$ based on $\underline{\lambda}_{1,1}^*$ and $\underline{\lambda}_{1,2}^*$,
- repeat the iteration procedure.

The above proofs only demonstrate that at each iteration, a new set of upper and lower bounds are obtained for boundary fluxes and boundary velocity potentials. The next step is to show that the above iterative scheme is stable. Also it is necessary to understand the important factors effecting the rate of convergence.

Computationally, the efficiency of the above scheme is based on the assumption that a large system can be divided into a series of blocks which can be individually stored in the main storage area of a computer. Blocks can be processed individually in this scheme while the calculation of the surface fluxes is done through a relaxation scheme for each surface without ever solving a large system of equations. These computational considerations will later be discussed in detail.

D. The Convergence Characteristics of the Iterative Scheme

To understand the convergence characteristics of the iterative scheme, let us again consider the simple two-block model.

Start the solution by assuming an initial flux vector at the boundary: $\underline{\lambda}_1^o$. Then, from equation (30), we can calculate the potentials as follows:

$$\underline{\Phi}_1^* = \underline{A}_1^{-1}\underline{F}_1 - \underline{A}_1^{-1}\underline{C}_{1,1}^t\underline{\lambda}_1^o$$

$$\underline{\Phi}_2^* = \underline{A}_2^{-1}\underline{F}_1 - \underline{A}_2^{-1}\underline{C}_{2,1}^t\underline{\lambda}_1^o \tag{44}$$

- Calculate the potential vectors on the boundary surface and average two vectors on the boundary by using an averaging vector α.

$$\underline{\Phi}_{1,1}^* = \underline{C}_{1,1}\ \underline{\Phi}_1^*$$

$$\underline{\Phi}_{2,1}^* = \underline{C}_{2,1}\ \underline{\Phi}_2^* \tag{45}$$

$$\underline{\Phi}_{B,1}^* = \alpha\underline{\Phi}_{1,1}^* + (1 - \alpha)\ \underline{\Phi}_{2,1}^*$$

- Calculate the boundary fluxes at the same boundary by solving the following pairs of equations for each block by specifying potentials on the boundary surface or constraints.

$$\underline{A}_1\ \underline{\Phi}_{1,t}^* + \underline{C}_{1,1}^t\underline{\lambda}_{1,1}^* = \underline{F}_1$$

$$\underline{C}_{1,1}\ \underline{\Phi}_{1,t}^* = \underline{\Phi}_{B,1}^* \tag{46}$$

and

$$\underline{A}_2\ \underline{\Phi}_{2,t}^* + \underline{C}_{2,1}\ \underline{\lambda}_{1,2}^* = \underline{F}_2$$

$$\underline{C}_{2,1}\ \underline{\Phi}_{2,t}^* = -\ \underline{\Phi}_{B,1}^* \tag{47}$$

In the above equation values of the right-hand-side vectors are known.

- Average $\underline{\lambda}_{1,1}^{*}$ and $\underline{\lambda}_{2,1}^{*}$ solutions at the boundaries with a second averaging parameter β

$$\lambda_1^1 = \beta\underline{\lambda}_{1,1}^{*} + (1 - \beta)\ \underline{\lambda}_{2,1}^{*} \tag{48}$$

By combining equations (44 - 48) we can write the following recurrence relationship:

$$\underline{\lambda}^{n+1} = \underline{C} + \underline{D}\underline{\lambda}^{n} \tag{49}$$

where the coefficient matrix $\underline{D}$ can be written as follows:

$$\underline{D} = [\beta\tilde{\underline{A}}_1 - (1 - \beta)\tilde{\underline{A}}_2]\ [\alpha\tilde{\underline{A}}_1^{-1} - (1 - \alpha)\tilde{\underline{A}}_2^{-1}] \tag{50}$$

Here $\tilde{\underline{A}}_{1,1}$ and $\tilde{\underline{A}}_{2,1}$ are reduced coefficient matrices related to the boundary surface in the following form:

$$\tilde{\underline{A}}_1 = \underline{C}_{1,1}\ \underline{A}_1\ \underline{C}_{1,1}^{t}$$

$$\tilde{\underline{A}}_2 = \underline{C}_{2,1}\ \underline{A}_2\ \underline{C}_{2,1}^{t} \tag{51}$$

The rate of convergence depends on the eigenvalues of the coefficient matrix $\underline{D}$, which can be simplified as,

$$\underline{D} = (1 - \alpha - \beta + 2\alpha\beta)\underline{I} - \beta(1 - \alpha)\tilde{\underline{A}}_1\tilde{\underline{A}}_2^{-1} - \alpha(1 - \beta)\tilde{\underline{A}}_1\tilde{\underline{A}}_1^{-1} \tag{52}$$

where $\underline{I}$ is the identity matrix. In the case of two equal size blocks with $\underline{A}_1 = \underline{A}_2$, $\underline{D}$ can be written as follows:

$$\underline{D} = (1 - 2\alpha - 2\beta + 4\alpha\beta)\underline{I}$$

In this case for $\alpha = \beta = 0.5$, an exact solution is obtained after one step.

E. Stability Limits for the Iterative Solution Scheme

To illustrate the bounds for convergence of the developed scheme, a simplified one-dimensional model will be utilized. In this case, the coefficient matrix becomes a scalar, d and $\tilde{\underline{A}}_1^{-1}\tilde{\underline{A}}_2$ matrix can be represented by another scalar r. Of course $\tilde{\underline{A}}_2^{-1}\tilde{\underline{A}}_1$, becomes 1/r. Then, the coefficient matrix in equation (52) can be written as

$$d = 1 - \alpha - \beta + 2\alpha\beta - \beta(1 - \alpha)r - \alpha(1 - \beta)/r \qquad (54)$$

The stability condition for this system, which requires

$$- 1 < d < 1 \qquad (55)$$

becomes

$$2 > \alpha + \beta - 2\alpha\beta + \beta(1 - \alpha)r + \alpha(1 - \beta)/r > 0 \quad . \qquad (56)$$

For example, $\alpha = \beta = 0.5$, this condition can be written as follows:

$$2 > 0.5 + 0.25r + 0.25/r > 0 \qquad (57)$$

Since r is positive for positive-definite systems, the stability condition requires

$$3 - \sqrt{8} < r < 3 + \sqrt{8} \qquad (58)$$

In a general case, r represents the ratios of eigenvalues of neighboring blocks.

As can be seen from the above expression, the relaxation scheme is convergent only for a narrow band of ratios of eigenvalues. This means that if the neighboring blocks are not similar in terms of shape, size or grid refinement, one cannot expect convergence from the iterative scheme. Of course in a general system, a series of eigenvalues are involved which we do not have prior knowledge of. The above results were obtained for $\alpha = \beta = 0.5$. A more general representation of the bands are shown in figure 5.

In order to improve the ratio of convergence, we introduced a relaxation parameter ω into the scheme in the following manner:

- Once the estimates for velocity potentials were calculated and an average was computed in equation (45), the velocity potentials for each surface vector was relaxed in the following manner:

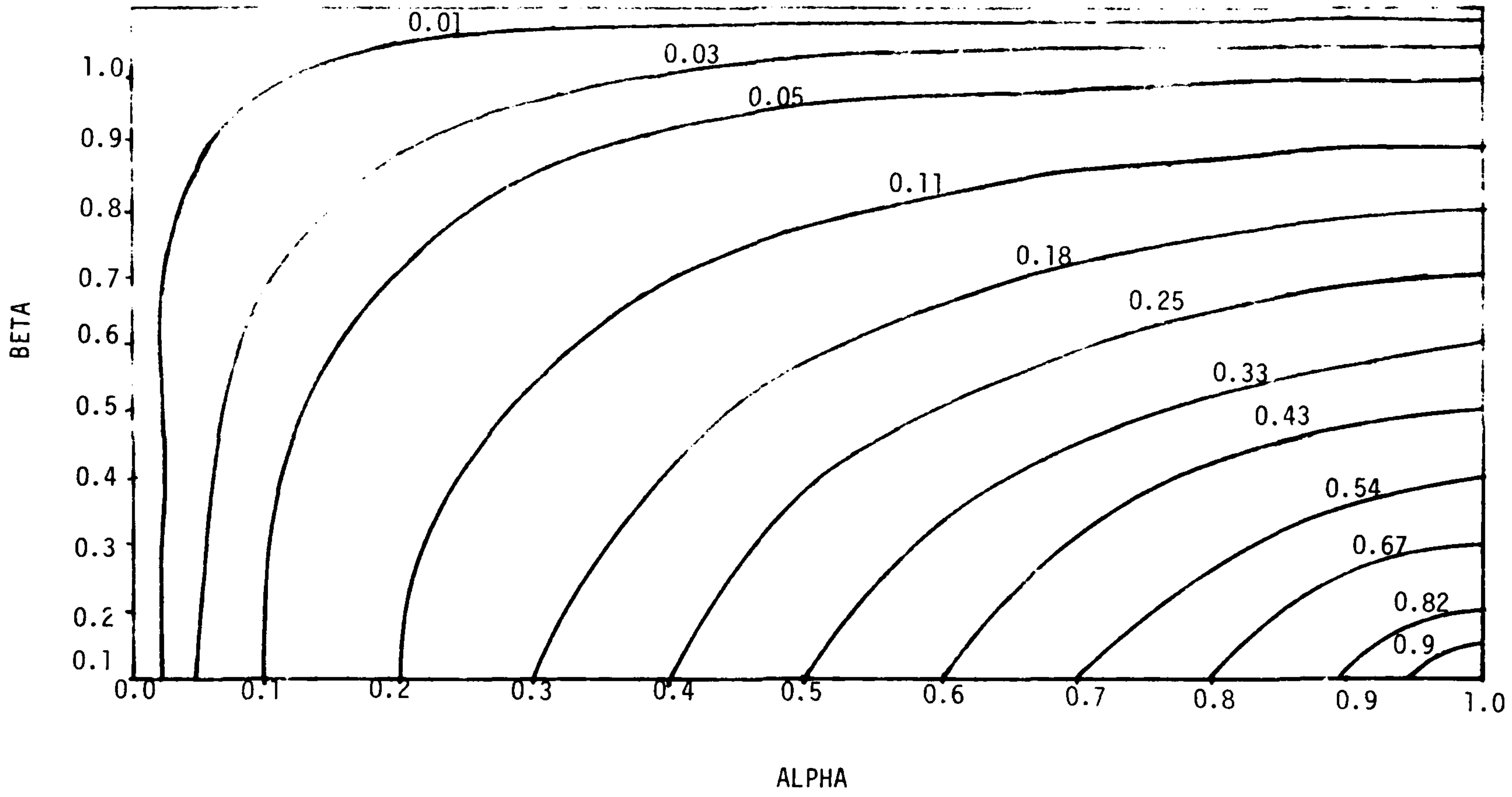


Figure 5a. Variation of the lower bound for r for given values of α and β.

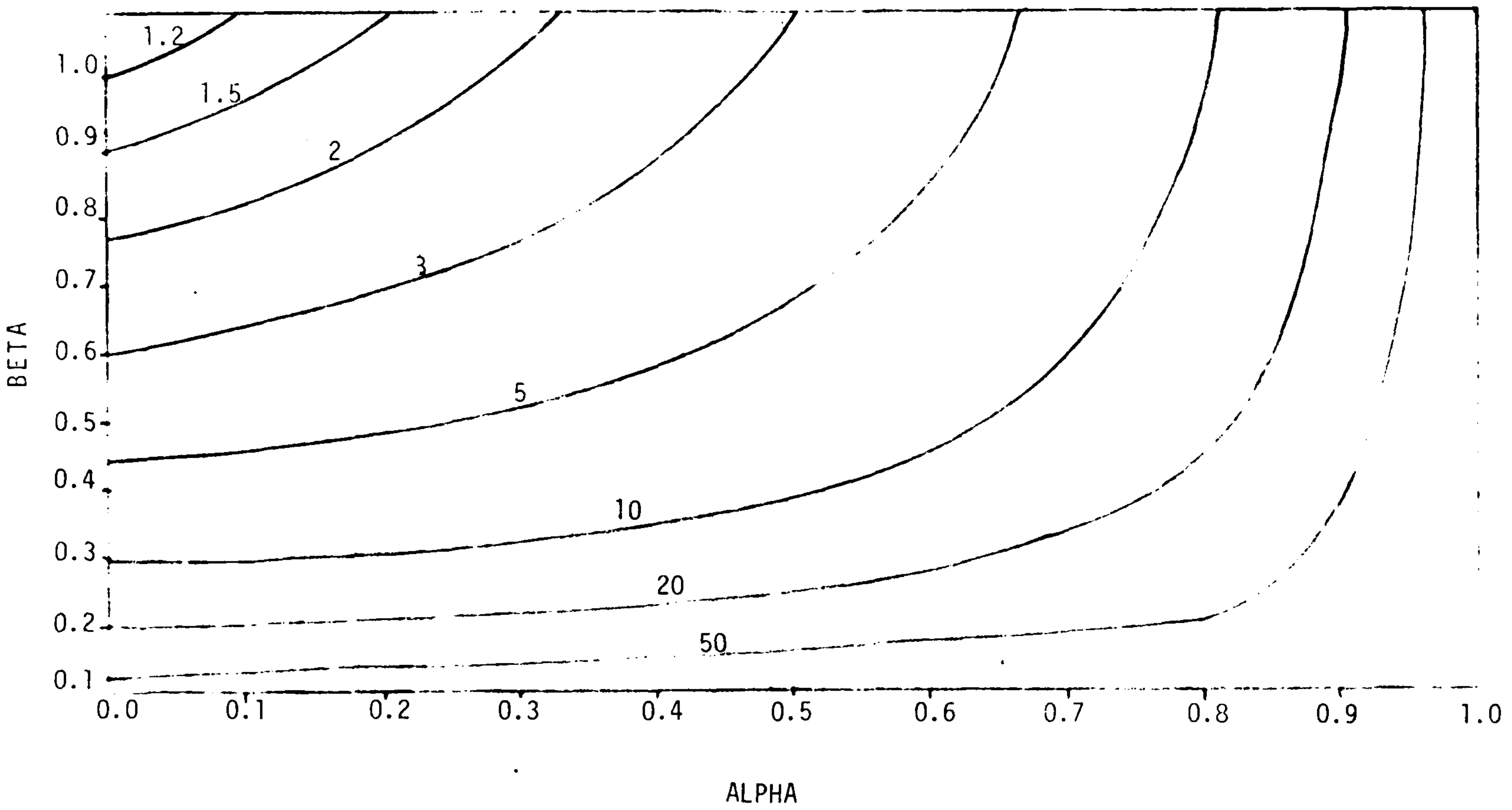


Figure 5b. Variation of the upper bound for r for given values of α and β.

$$\tilde{\underline{\Phi}}_{1,1} = \omega\tilde{\underline{\Phi}}_{B,1} + (1 - \omega)\ \underline{\Phi}^*_{1,1}$$

$$\tilde{\underline{\Phi}}_{2,1} - \omega\tilde{\underline{\Phi}}_{B,2} + (1 - \omega)\ \underline{\Phi}^*_{2,1} \qquad (59)$$

- The second part of equations (46) and (47) were then modified as follows:

$$\underline{C}_{1,1}\ \underline{\Phi}^*_{1,t} = \tilde{\underline{\Phi}}_{1,1}$$

$$\underline{C}_{2,1}\ \underline{\Phi}_{2,t} = -\ \tilde{\underline{\Phi}}_{2,1} \qquad (60)$$

- An average for $\underline{\lambda}$ vector is again calculated from equation (48). However, in this case, the $\underline{\lambda}$ vector is relaxed again with the same relaxation factor to calculate the fluxes for each surface as follows:

$$\underline{\lambda}_{1,1} = \omega\underline{\lambda}_1^1 + (1 - \omega)\underline{\lambda}^*_{1,1} \qquad \underline{\lambda}_{1,2} = \omega\underline{\lambda}_1^1 + (1 - \omega)\underline{\lambda}^*_{2,1} \qquad (61)$$

In this case the coefficient matrix $\underline{D}$ can be written in the following form:

$$\underline{D} = [1 - \omega\alpha - \omega\beta + 2\omega\alpha\beta]\underline{I} - \beta\omega(1 - \alpha)\underline{A}_1\underline{A}_2^{-1} - \omega\alpha(1 - \beta)\underline{A}_2\underline{A}_1^{-1} \qquad (62)$$

The stability condition for the one-dimensional case requires that

$$2 > \omega\ [\alpha + \beta - 2\alpha\beta + \beta(1 - \alpha)r + \alpha(1 - \beta)/r] > 0 \qquad (63)$$

In comparison with the stability condition in equation (56), in this case ω becomes the controlling factor. Figure 6 illustrates the bounds for stability with different relaxation parameters ω for a fixed value of $\alpha = \beta = 0.5$. In the actual computations, as it will be discussed later, both were fixed at 0.5 and the stability was controlled by only changing ω.

For a general problem, r becomes the ratio of the largest eigenvalue for one block to the smallest eigenvalue of the neighboring blocks or vice versa. The magnitude of the largest eigenvalue of the coefficient matrix for a block depends on the size of the smallest element. On the other

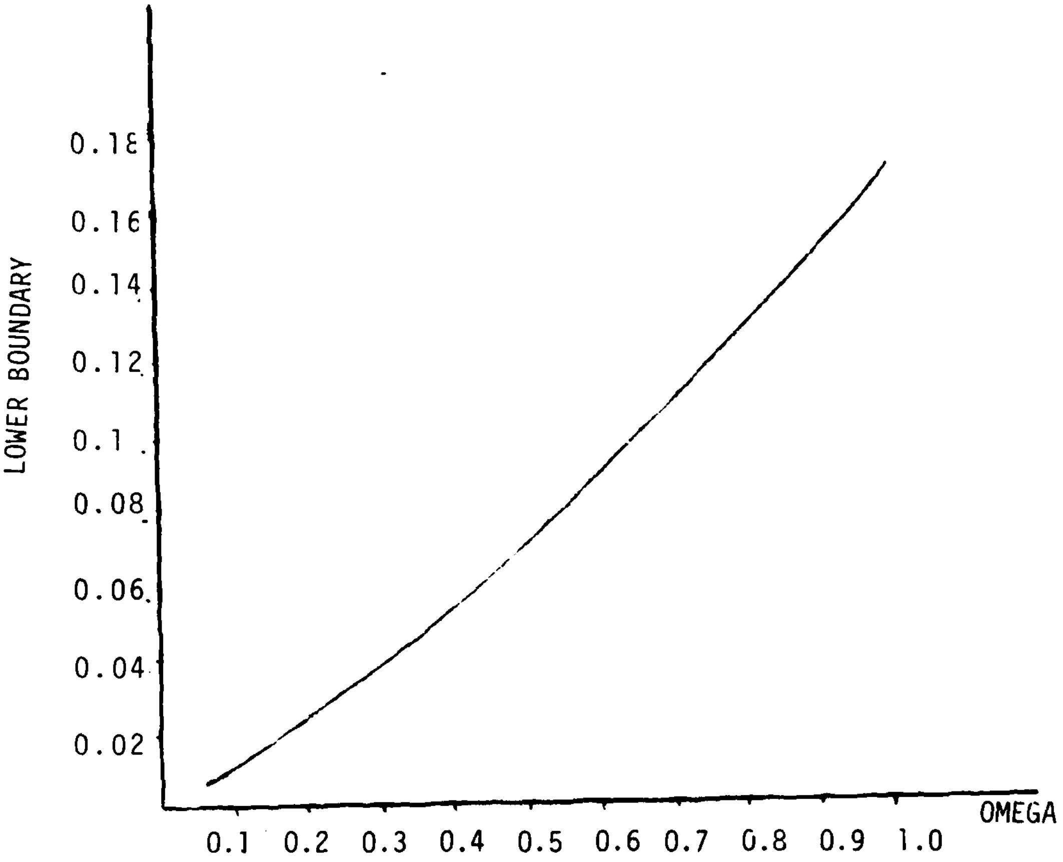


Figure 6a. Variation of lower bound for r with relaxation parameter (ω). ($\alpha=\beta=0.5$)

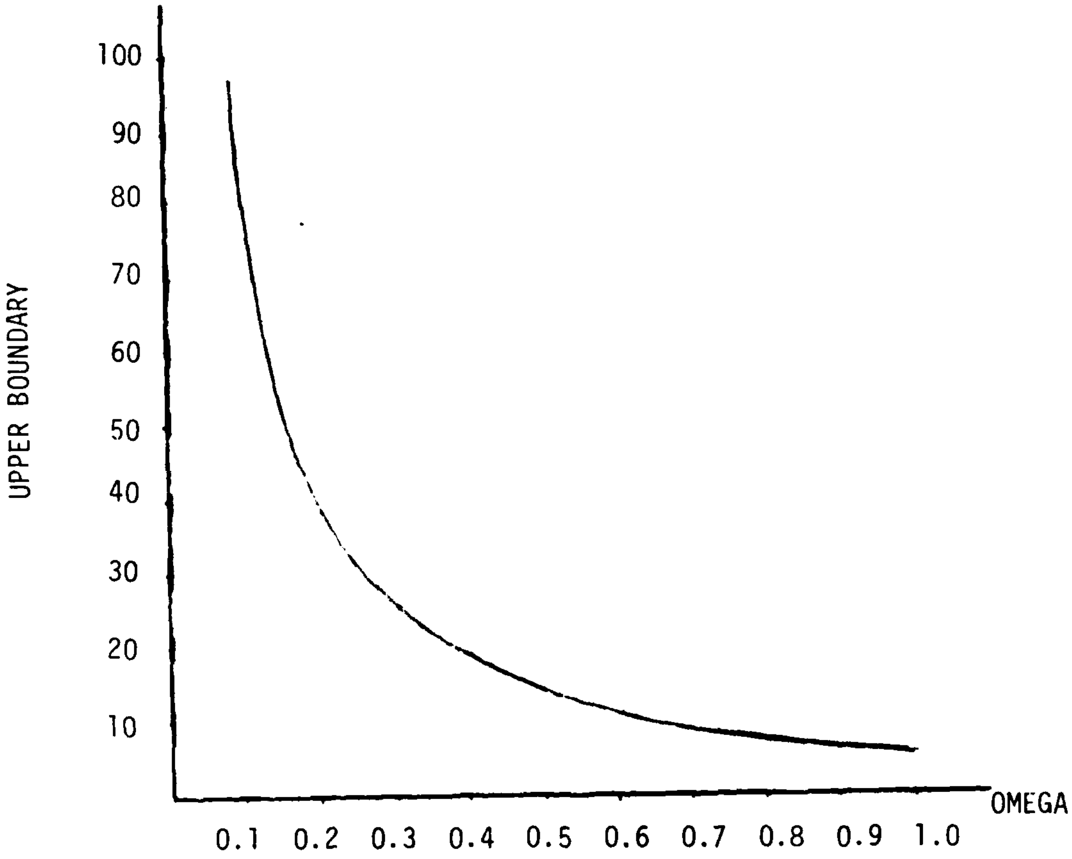


Figure 6b. Variation of upper bound for r with relaxation parameter (ω). ($\alpha=\beta=0.5$)

hand, the smallest eigenvalue becomes lower with the increasing size of a block. Thus, we expect small relocation parameters to be employed when large blocks are connected with small blocks with excessive grid refinements.

It remains an interesting problem to experiment with different values α and β for each block interface since there is no requirement for α and β to retain constant values during the iteration scheme.

III. COMPUTATIONAL PROCEDURE

In the above discussions, we tried to summarize the fundamental principles and objectives in designing the present computational scheme. In this section, we will try to describe the numerical procedure in detail, emphasizing its computational aspects. A flow-chart summarizing the iterative scheme is shown in figure 7. In this flow-chart, block surface means one surface of a block attached to another single neighboring block. A simple example for a two-block structure is shown in Figure 8.

As can be seen from this flow-chart, the first step is to initialize the density of each element in all blocks and assign a boundary flux distribution for all global boundaries and inter-block boundaries as defined in equations (11-12). In actual computations, we may initialize the problem, for example, by setting the densities to zero everywhere and by assuming the flux boundary conditions at the upstream to be valid for all block boundaries.

We then perform the first set of operations. We consider each of the blocks one at a time. There is no order to the sequence in which the blocks are operated on. Given a series of processors, these block operations can be distributed among these processors. For each block we performed what we called operation A, as described in figure 9. This operation involved the solution of the conservation of mass equation for each block individually under the set of given boundary fluxes as specified in equations (11-12).

One problem in this operation, which was mentioned previously, was the necessity to specify the velocity potential at one grid point due to the singularity of the system. It was pointed out that this value could

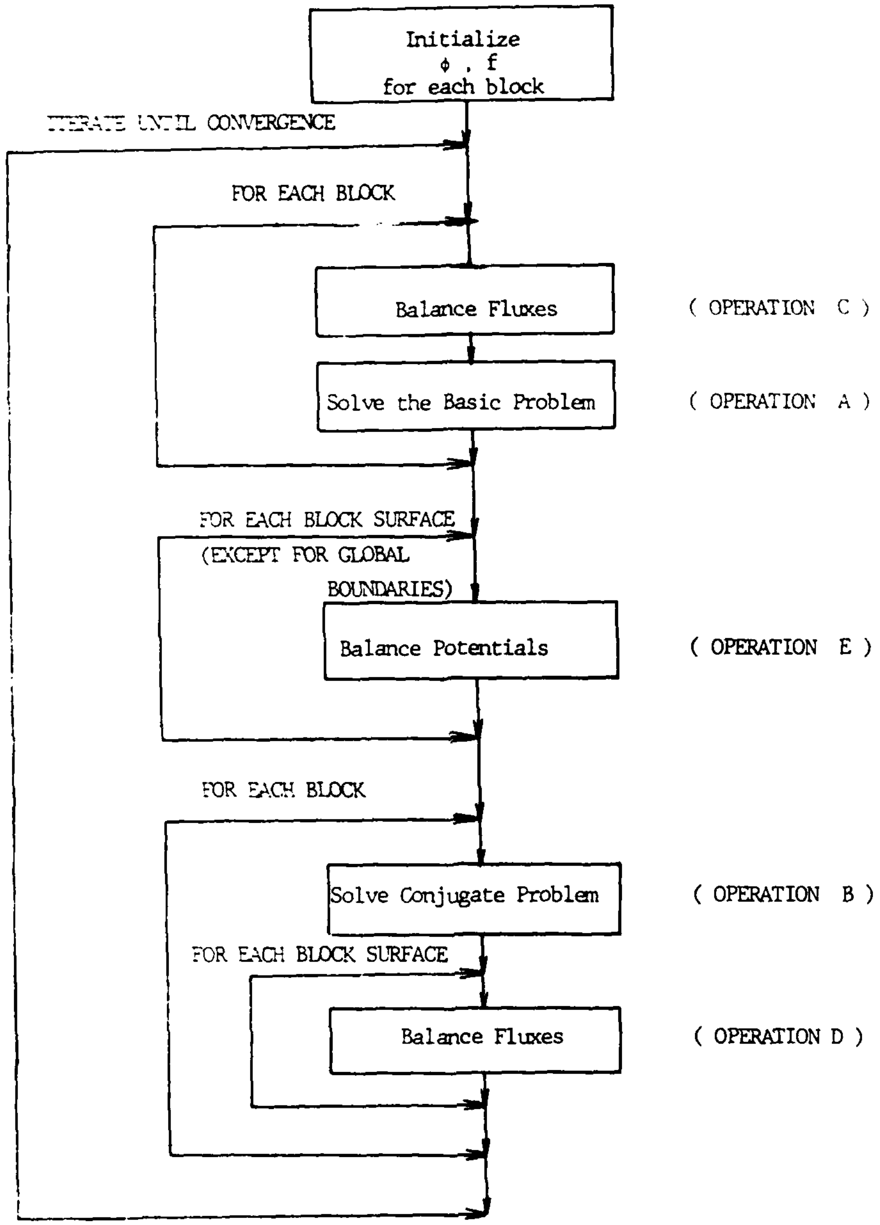


Figure 7. Flowchart of the Block-Structured Solution Scheme.

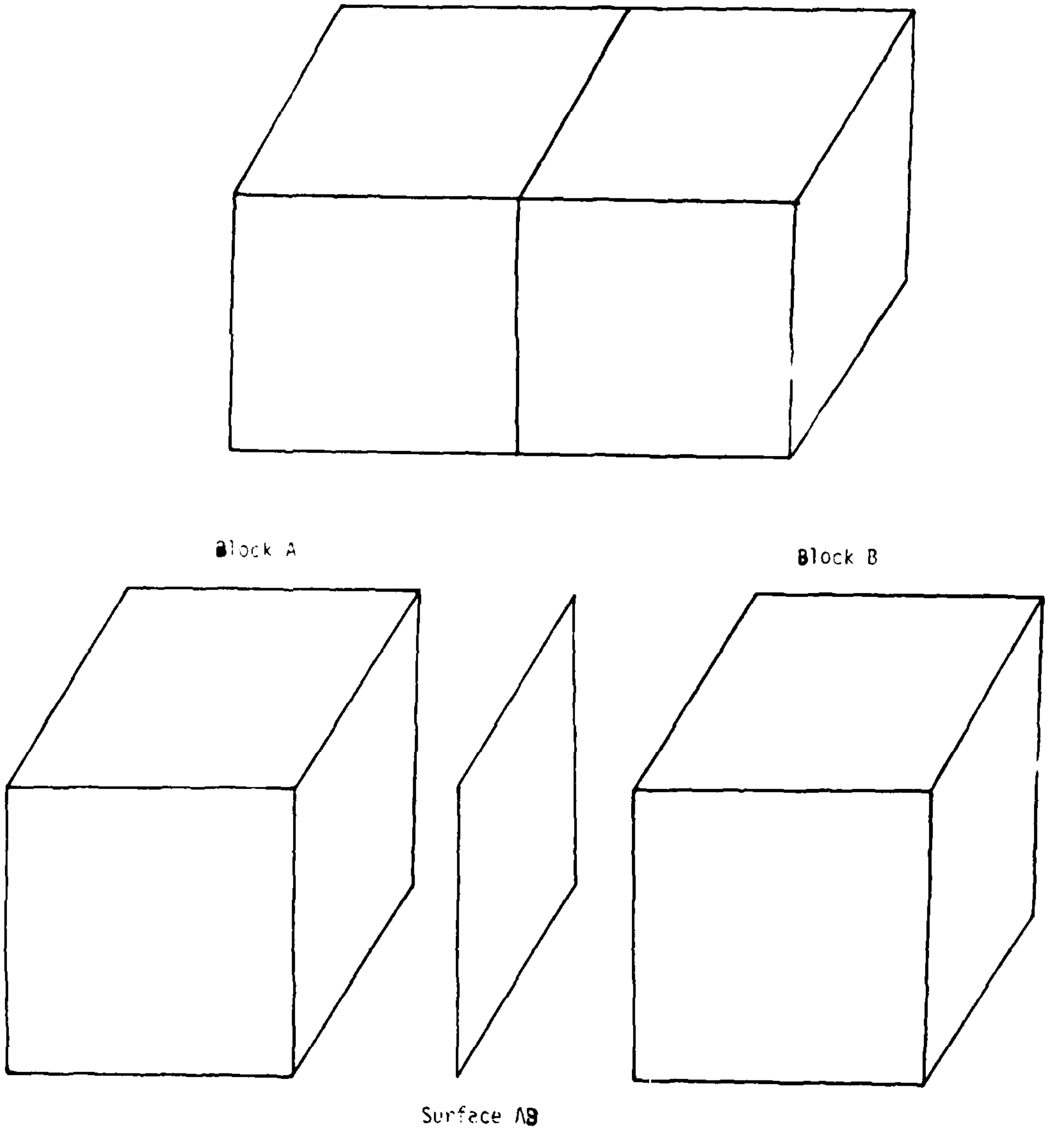
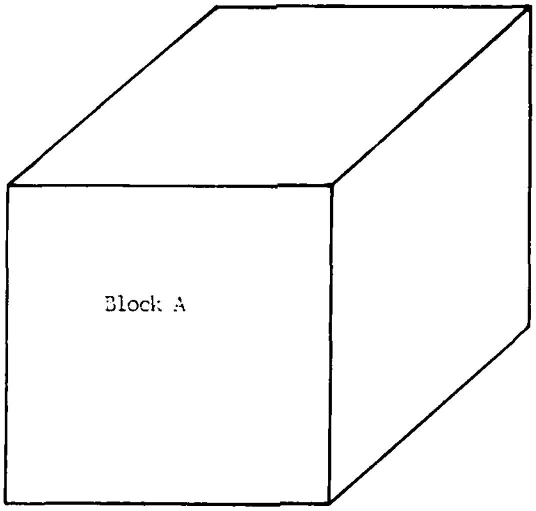


Figure 8. A Two-Block Structure Describing the Basic Elements of the Solution Scheme.



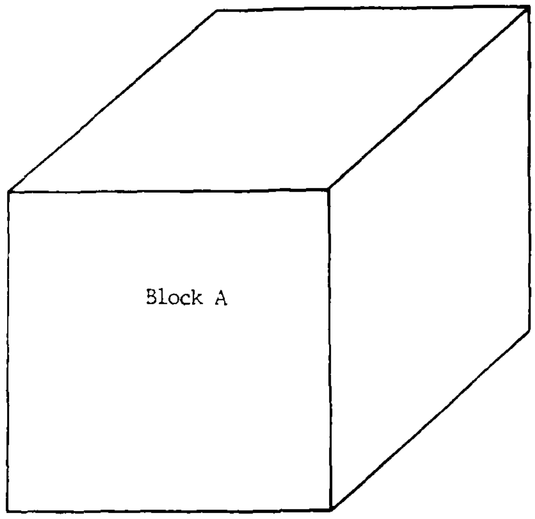
$$
\begin{aligned}
\underline{\nabla}\cdot(\rho\underline{\nabla}\phi) &= 0 \\
\rho\,\underline{n}\cdot\underline{\nabla}\phi &= f
\end{aligned}
$$

Figure 9. Operation A: Definition of the Basic Problem for a Block.

be arbitrarily chosen if the conservation of mass for the entire block was satisfied as stated in equation (14). For this purpose, we first scaled the mass fluxes over the entire boundary which was called operation C in the flow-chart as shown in figure 10. Details of this scaling process will be discussed later. After the operation A was performed, a set of velocity potentials was obtained and stored for each block ($\underline{\Phi}_1^*$ in equation (31)) as individual sets.

The second set of operations included the treatment of each of the surfaces. This was called operation E in the flow-chart and is illustrated in figure 11. In this case, for each surface, velocity potentials, calculated from individual blocks, were picked, averaged by using equation (46) and relaxed as defined in equation (60). Again each surface was addressed and processed individually. However, a certain restriction existed here which was due to the fact that a block may have several surfaces. When processing a particular surface, data for a particular block may not be readilyavailable in a parallel processing environment since it could be addressed at the same time by another surface.

The last loop was again a block operation. This was called operation B, as shown in figure 12. For each block, we picked the averaged and relaxed values of surface potentials. We imposed these values on boundary conditions as given in equations (39) and solved the conservation of mass equation one more time but with this new boundary condition. The outcome of this operation was the boundary fluxes for each block. This information was written on a surface file again to be averaged and relaxed. Thus, each time a block was processed, its contribution to the flux balancinq (operation D, shown in figure 13) was simultaneously performed as indicated in the flow-chart.

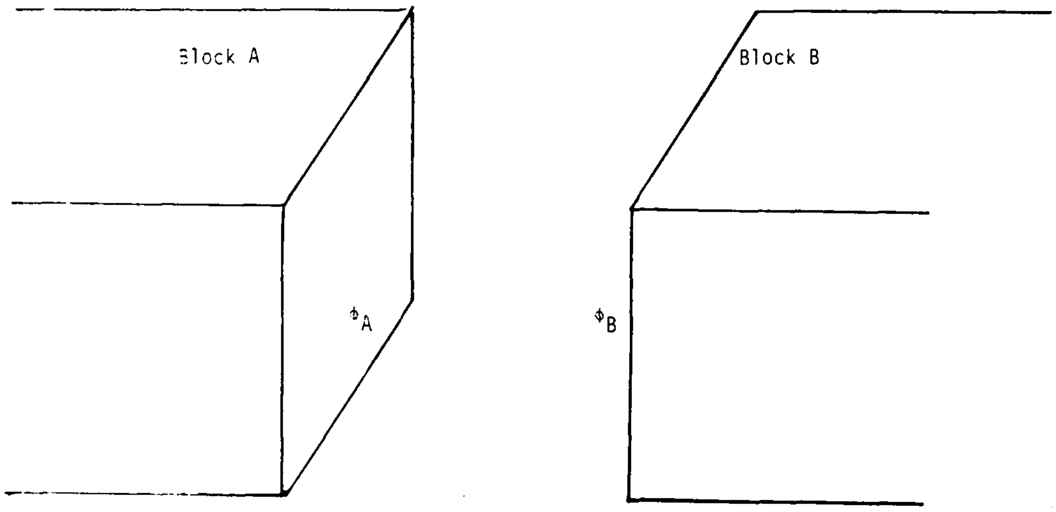


$$\int_s f\,ds = \varepsilon$$

$$\int_s |f|ds = Q$$

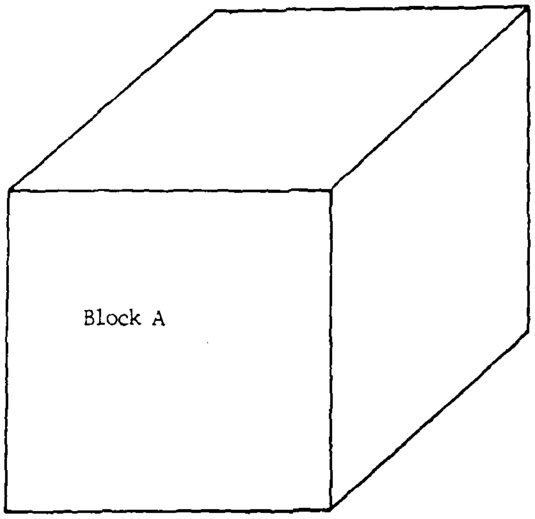
$$\hat{f} = f - \frac{\varepsilon}{Q}|f|$$

Figure 10. Operation C: Balancing of Block Fluxes.



$$\hat{\phi} = \omega\phi_A + (1-\omega)\phi_B$$

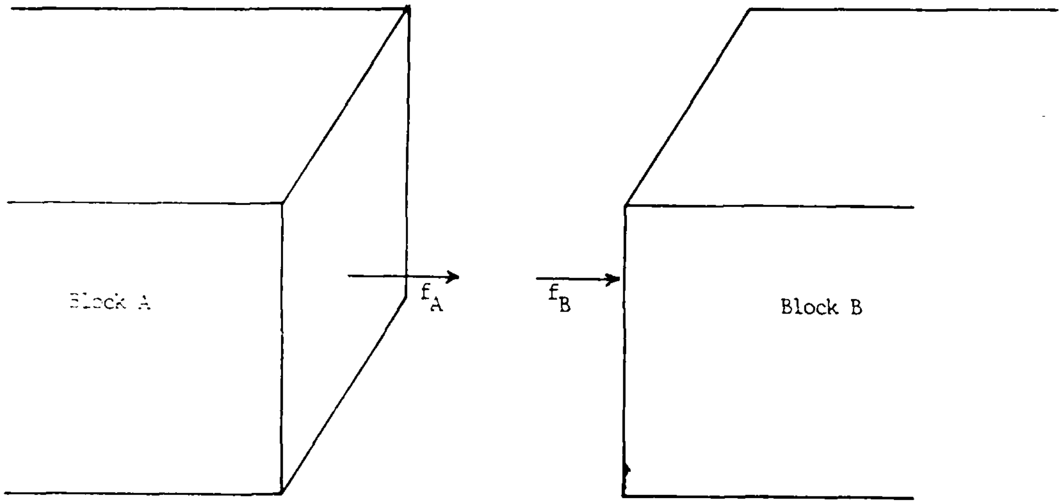
Figure 11. Operation E: Balancing Potentials for Two Blocks.



$$\underline{\nabla}\cdot(\rho\underline{\nabla}\phi) = 0$$

$$\phi = g$$

Figure 12. Operation B: Definition of the Conjugate Problem.



$$\hat{f} = \omega f_A + (1-\omega) f_B$$

Figure 13. Operation D: Flux Balancing on Block Boundary Surfaces.

The details of these operations are listed below.

A. Operation A

Operation A involves the iterative solution of the potential flow equation for each block as defined in equations (10-12). We employ an iterative procedure for each block as summarized below:

At each iteration, equation (18) has to be solved to determine the nodal values of the velocity potentials. Rather than solving equation (18) directly, we employ the following iterative procedure:

$$\underline{A}_i^o \underline{\Phi}_i^{n+1} = \underline{F}_i - \underline{C}_{i,j}^t \underline{\lambda}_j^n - \underline{A}_i^n \underline{\Phi}_i^n \qquad (64)$$

Here n indicates the iteration step, where $\underline{\Phi}$ and $\underline{\lambda}$ vectors are calculated from the previous iteration. Coefficient matrix $\underline{A}_i^n$ is determined by using the densities obtained from the previous iteration step. $\underline{A}_i^o$ is calculated at the first step by employing a fixed density distribution.

The right-hand-side vector is calculated for each element of a block and assembled as a block residual vector. Matrix $\underline{A}_i^n$ is never assembled but calculated element-by-element to be multiplied by a vector consisting of the nodal values of element velocity potentials. The left-hand-side coefficient matrix is a symmetric, positive-definite matrix which can be decomposed only once and stored. In this case, each time a residual vector is determined for a vector, a forward-backward substitution operation is performed on this coefficient matrix. The size of the coefficient matrix is equal to the number of the grid points in a block times the half of the wavefront. For example, for a regular grid of 10 x 20 x 30 grid points, the wavefront would be equal to 10 x 20 x 2 = 80. If sufficient core is not available to store such information, the coefficient matrix has to be assembled and decomposed at each iteration step.

In our computations, considerable time was spent in terms of formulating the residual vector. This computational time can be reduced considerably if the matrices in equations (19-22) can be stored and read-in at every iteration step. Only the density has to be recalculated at each iteration. In this case, the core requirements increase by approximately 84 x number of elements for each block.

The details of the above iteration scheme have been previously presented in detail [12]. Its convergence rate depends strongly on the manner artificial density is included in the computations. The same scheme has been applied without any modification to the block-structured scheme. As it will be shown later, no significant reduction in the rate of convergence was observed after the application of the block-structured solution scheme as compared to solving the same problem as a single block.

B. Operation B

Operation B involves the solution of conservation of mass equation for a slightly different problem. In this case, the velocity potentials at the inter-block boundaries are specified and the normal mass fluxes at the same boundaries have to be determined. This problem can be defined as follows: Consider the solution of the conservation of mass equation,

$$\underline{\nabla}\cdot(\rho\underline{\nabla}\phi) = 0 \qquad \text{in } \Omega$$

with

$$\rho\underline{u}\cdot n = f \qquad \text{on } \Gamma^{o}$$

and

$$\phi = p \qquad \text{on } \Gamma^{c} \tag{65}$$

The variational problem can then be written as follows:

$$\pi = \sum_{i}\int_{\Omega_i}\frac{1}{2}\rho\underline{\nabla}\phi_i\cdot\underline{\nabla}\phi_i \; d\Omega + \int_{\Gamma_2^o} \phi_i f_i \; d\Gamma + \int_{\Gamma_2^c} \gamma_i(\Phi_i - p) \; d\Gamma \tag{66}$$

In this case, the Lagrange multiplier γ_i on the boundary surface Γ_c indicates the unknown boundary flux to be determined which corresponds to the known velocity potential distribution as defined by p(s,t).

After substituting the finite element approximation, the governing equations can be written as follows:

$$\begin{bmatrix} \underline{A}_i & \underline{D}_{ij} \\ \underline{D}^t_{i,j} & 0 \end{bmatrix} \begin{bmatrix} \underline{\Phi}_i \\ \underline{\gamma}_{i,j} \end{bmatrix} = \begin{bmatrix} \underline{F}_i \\ \underline{p}_j \end{bmatrix} \qquad (67)$$

Here, $\underline{\gamma}_{i,j}$ vector represents the unknown mass fluxes on each surface grid point that is connected to a neighboring block surface. The solution of equation (67) can be written as follows:

$$\underline{\Phi}_{i,j} = (\underline{D}^t_{i,j}\underline{A}^{-1}_i\underline{D}_{i,j})^{-1}\ [\underline{D}^t_{i,j}\underline{A}_i\underline{F}_i - \underline{p}_j] \qquad (68)$$

where i is the block number and j is the surface number.

As can be seen from equation (68), the mass fluxes at the inter-block boundaries can be written in terms of mass flux vector at global boundaries ($\underline{F}_i$) and the known values of the velocity potential vector at the inter-block boundaries ($\underline{p}_i$). The first part is a constant vector through the iterations while, $\underline{p}_j$ vector is recalculated at each iteration. The coefficient matrix, in equation (68), is simply the reduced form of the matrix $\underline{A}_i$ where all of the grid points on the boundaries are eliminated. This coefficient matrix is different then the one employed in Operation A. Again by using a constant coefficient matrix scheme, $\underline{D}$ can be calculated only once and stored. For a block of 10 x 20 x 30 grid points it will have a size of 8 x 18 x 28. It will also have a wavefront of 2 x 8 x 18.

There is a basic difficulty in the above computational scheme. Let us assume that a block has six surfaces and all of these six surfaces are attached to neighboring blocks. In this case, the $\underline{p}_j$ vector has to be calculated from all six surfaces. However, there is no restriction to ensure that the velocity potentials calculated from each surface are continuous along the edges connecting such surfaces. A unique value cannot be assigned to $\underline{\Phi}$ to form the $\underline{p}_j$ vector at these points. Thus, the above set of computations in equation (68) has to be performed for each surface individually.

C. Operation C

Consider a single block with a series of boundary fluxes on each surface. For a rectangular block this would mean six individual surfaces. In the present scheme, all of these fluxes are calculated individually through computations involving the neighboring blocks. We also know that we can solve the conservation of mass equation only if it is satisfied globally for the entire block. To ensure such a condition, the total mass flux is calculated from equations (11) and (12) as follows:

$$\int_{\Gamma_o} f \, d\Gamma + \int_{\Gamma^c} g \, d\Gamma = \varepsilon \qquad (69)$$

Since the boundary conditions on the global boundary is exact, the error can be distributed to the other boundary surfaces in the following manner:

$$\int_{\Gamma^c} |g| \, d\Gamma = Q$$

$$\hat{g} = g - \frac{\varepsilon}{Q} |g| \qquad (70)$$

This process ensures that the conservation of mass is always satisfied before Operation A is performed.

D. Operation D

Operation D involves the balancing of fluxes as calculated from neighboring blocks. As indicated in equation (64), at the end of

Operation B, a set of boundary fluxes are calculated on each surface. In general, we can describe such surfaces with a set of discrete points. Let us assume that block $\underline{A}_1$ which is joined to this particular surface has m_1 grid points while block $\underline{A}_2$ attached to the same surface by m_2 grid points. Let us also assume that $m_1 \neq m_2 \neq n$ and all three sets do not physically match in space with each other.

By using a finite element procedure, we can employ a shape function as defined in equation (17) for each surface in the following form:

$$g_j(s,t) = \tilde{N}_k(s,t)\lambda_{i,k}(k=1,n)$$

$$g_{i,j}(s,t) = \tilde{N}_k(s,t)\lambda_{i,j,k}(k=1,m_1)$$

$$g_{m,j}(s,t) = \tilde{N}_k(s,t)\lambda_{m,j,k}(k=1,m_2) \qquad (71)$$

Where $\lambda_{i,k}$ represents the nodal fluxes on the surface nodes, $\lambda_{i,j,k}$ and $\lambda_{m,j,k}$ the nodal fluxes on the surface nodes of the neighboring blocks i and m. Then, by following the averaging scheme in equation (49) and the relaxation scheme in equation (61), it is possible to evaluate

$$\underline{g}_j^{n+1} = \underline{p}\underline{g}_{i,j} + (1 - p)\underline{g}_{m,j}$$

$$\underline{g}_{i,j}^{n+1} = \omega\underline{g}_i^{n+1} + (1 - \omega)\underline{g}_{i,j}^{n}$$

$$\underline{g}_{m,j}^{n+1} = \omega\underline{g}_j^{n+1} + (1 - \omega)\underline{g}_{m,j}^{n} \qquad (72)$$

In the case of blocks with matching grid points on the neighboring surfaces, the above operation is conducted directly on the nodal values of the $\underline{\lambda}$ vector.

It should be mentioned that the above operation is in fact a block based operation. For each element of the block, $\underline{\lambda}$ vector is calculated for each surface and stored on a surface based file according to the equation (72). After each block is processed, the outcome is the $g_{i,j}$ or $g_{m,j}$ vector on each surface. The computational cost of the operation is based on the effort in accessing such a surface file. For example, when one block is processed, the computed surface fluxes has to be written on each of the surfaces attached to that block. The same surface will then be accessed by the two blocks it is attached to. If a surface information file is kept in main memory, which includes both surface fluxes and geometry of the elements on both side of this surface, all of the above information exchange can be directly made in the main memory. Otherwise, an auxiliary file has to be accessed for storing such information. Knowing the connectivity of the blocks and the sequence in which this information has to be accessed, one can optimize such an operation for directing large systems.

E. Operation E

Operation E is very similar to Operation D but somewhat more direct. Velocity potentials are calculated from each surface, averaged and relaxed. Again a block based information file is generated which involves the velocity potentials for each of the grid points for a block. At each step, the surface potentials are updated once.

F. File Structure of the Operation

Two types of basic information are necessary for the operation: Velocity potentials and boundary fluxes. This information is stored for both Operations C and D in the following manner:

1. Block file

This file includes the following information for each block:

- Velocity potentials for each node
- Geometry of all the elements
- Shape functions for all of the elements
- Element coefficient matrices in equations (19 - 22)
- Decomposed coefficient matrices for a block.

This file is read in for one block at a time. All blocks can be processed sequentially or in any order.

2. Surface file

This file includes the following information for each surface:

- Surface fluxes for elements on both sides of the surface
- Geometry of the elements on both sides of the surface
- Shape functions for these elements
- Element matrices for these elements.

If we review the flow-chart, the following I/O operations are observed: Operations A and C require access to the block file while Operations B and D require access to the surface file.

In the application of the method on an IBM 4381 computer with 12 megabytes memory, the block and surface files were split into smaller files. Velocity potentials and boundary flux vectors were kept in main memory at all times, while the other information was read in when a particular block or surface was processed. This way most of the information transfer to the neighboring blocks in terms of averaging and relaxing was performed in the main memory.

IV. DISCUSSION OF RESULTS

To illustrate the applicability of the developed numerical procedure, a series of test cases was analyzed. As discussed in the previous sections, the convergence rate and bounds of convergence for this scheme depends on the size of the blocks and the grid refinement in each block. One can expect variations on the size of different blocks and the necessary grid refinement for each block in the case of complex three-dimensional flows. To test the robustness of the scheme, blocks were chosen in some cases in a way which was considered to be least favorable for convergence of the scheme. The test cases include several two-dimensional cases analyzed using the developed problem as well as a fully three-dimensional case as follows:

A. Incompressible flow around a NACA0012 airfoil

B. Compressible, subsonic flow around a NACA0012 airfoil ($M_\infty = 0.6$)

C. Transonic flow around a NACA0012 airfoil ($M_\infty = 0.83$)

D. Transonic flow around a wing-body configuration ($M_\infty = 0.90$, $\alpha = 3^o$)

A. Analysis of Incompressible Flow Around a NACA0012 Airfoil ($M_{in} = 0.001$)

Two types of block structures were employed for the analysis of this test case. The first case involved a one-dimensional block distribution in the flow direction with three blocks and two intermediate surfaces. This block structure is shown in Figure 14a. The number of grid points employed in each of the blocks are also shown in this figure. The generated grid for this block structure can be seen in Figures 14b - 14d. The second type of block structure, shown in Figure 15, involved twelve blocks and

No. of Grid Points

Incompressible flow	8x 7x3	10x 7x3	7x 7x3
Transonic flow	21x16x3	31x16x3	21x16x3

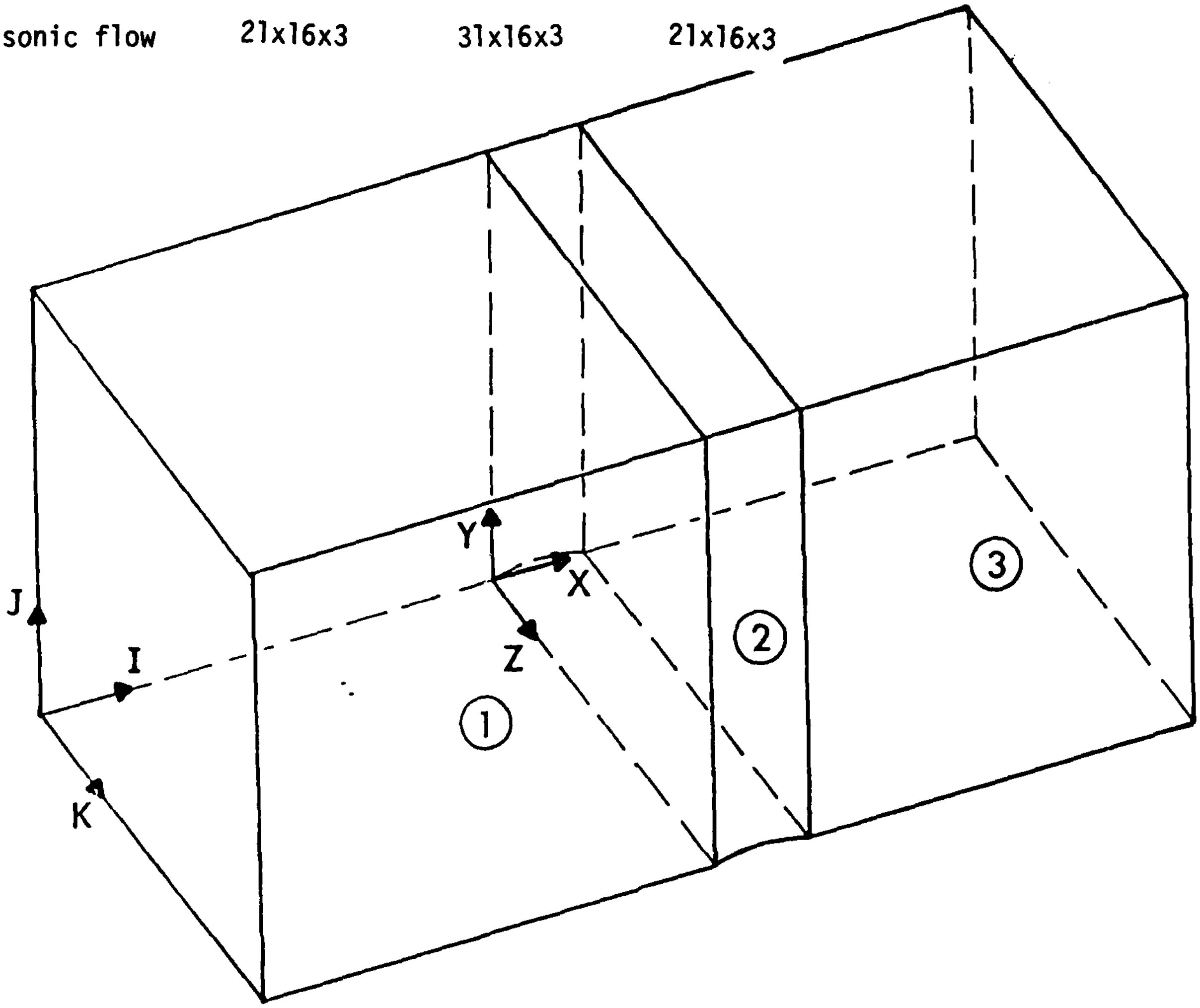


Figure 14a. Three-block grid structure for the analysis of flow around a NACA0012 profile.

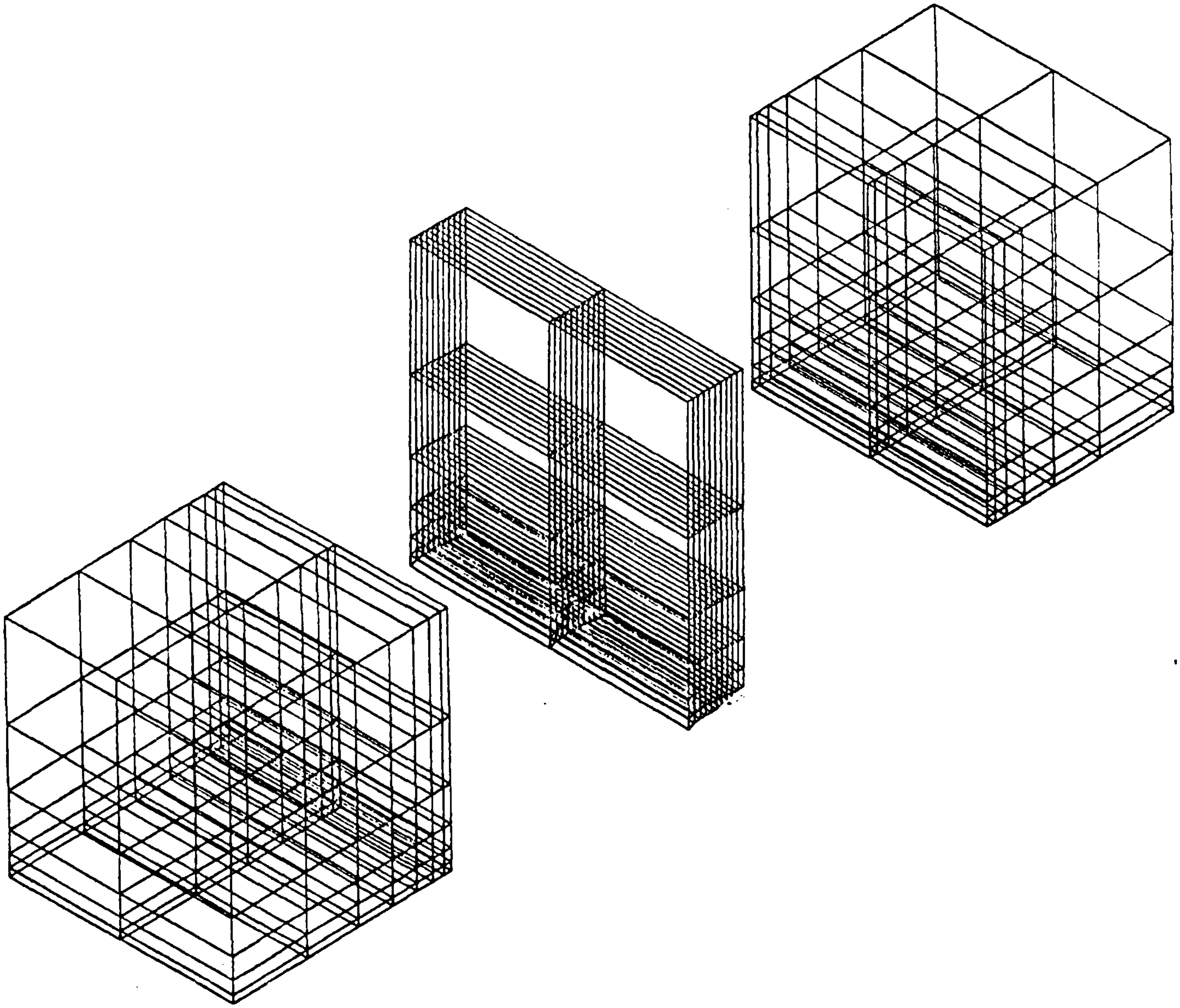


Figure 14b. Computational grids (3-block) for the analysis of flow around a NACA0012 profile.

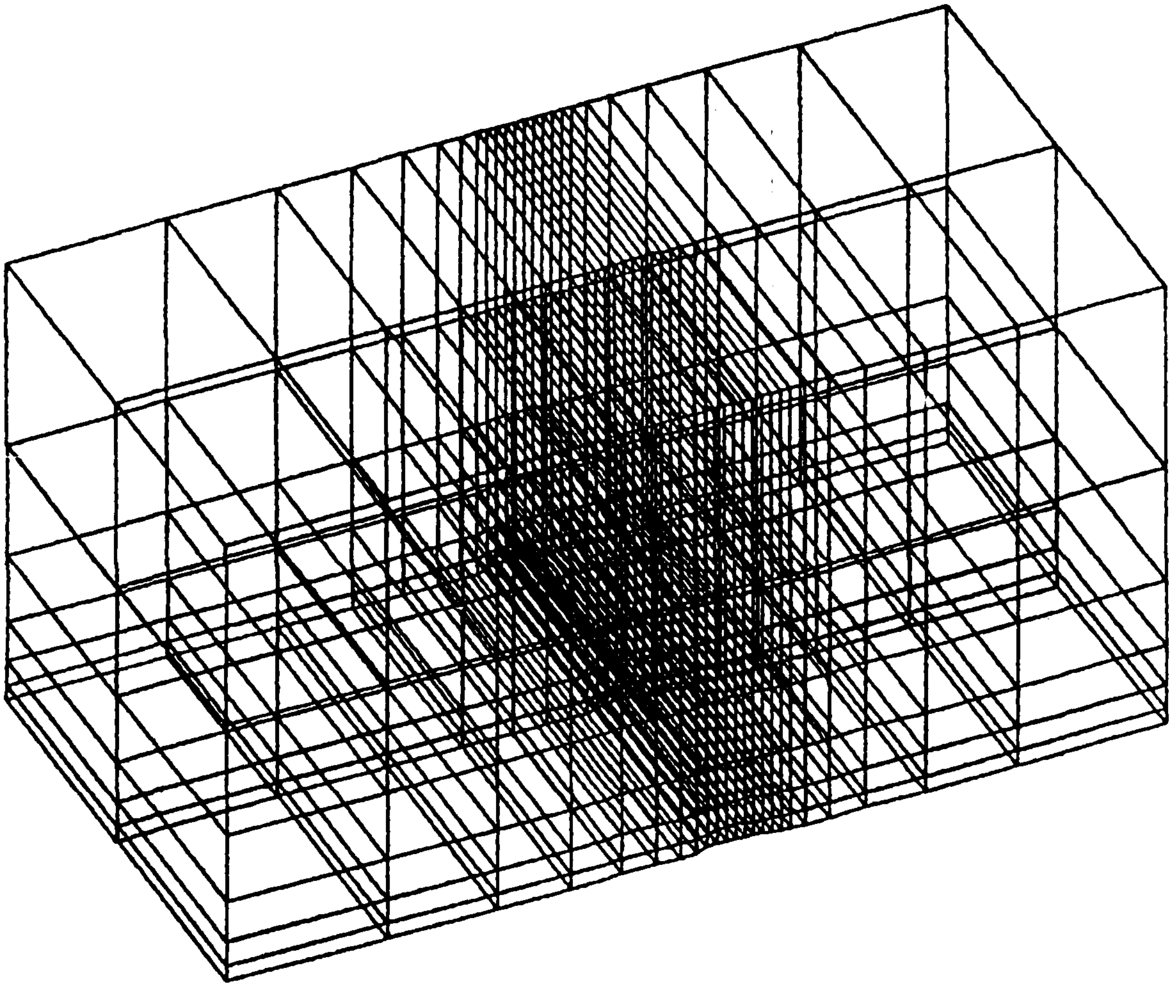


Figure 14c. Assembled computational grids (3-block) for the analysis of flow around a NACA0012 profile.

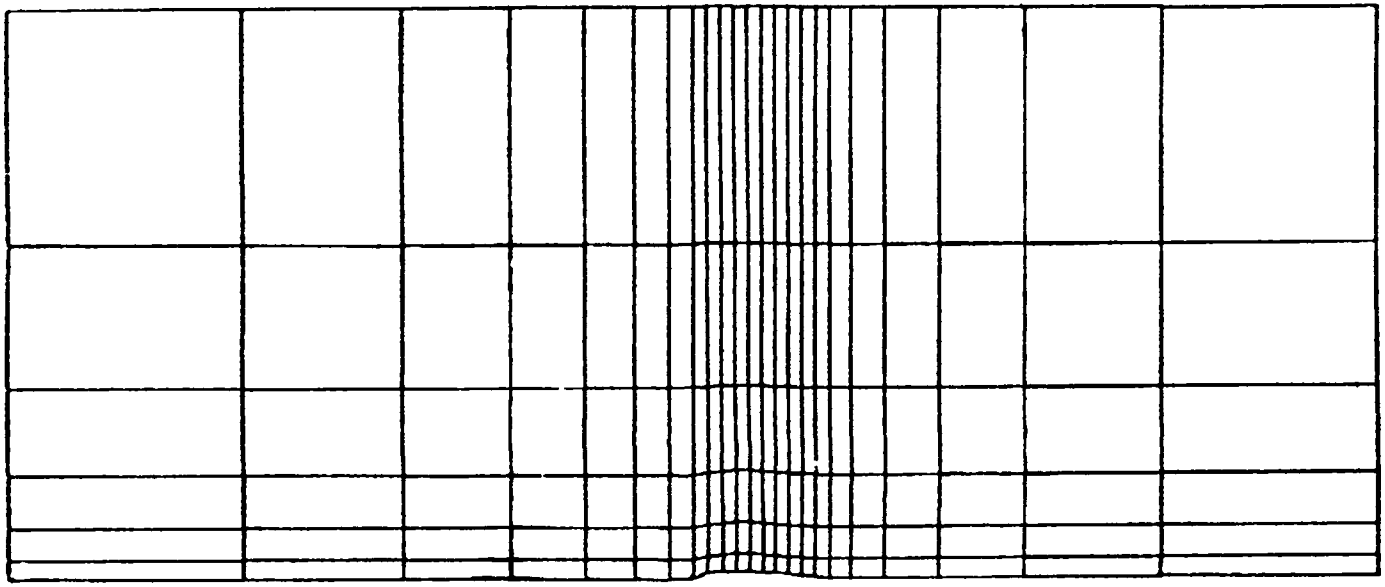


Figure 14d. A section of the 3-block grid for the analysis of flow around a NACA0012 profile.

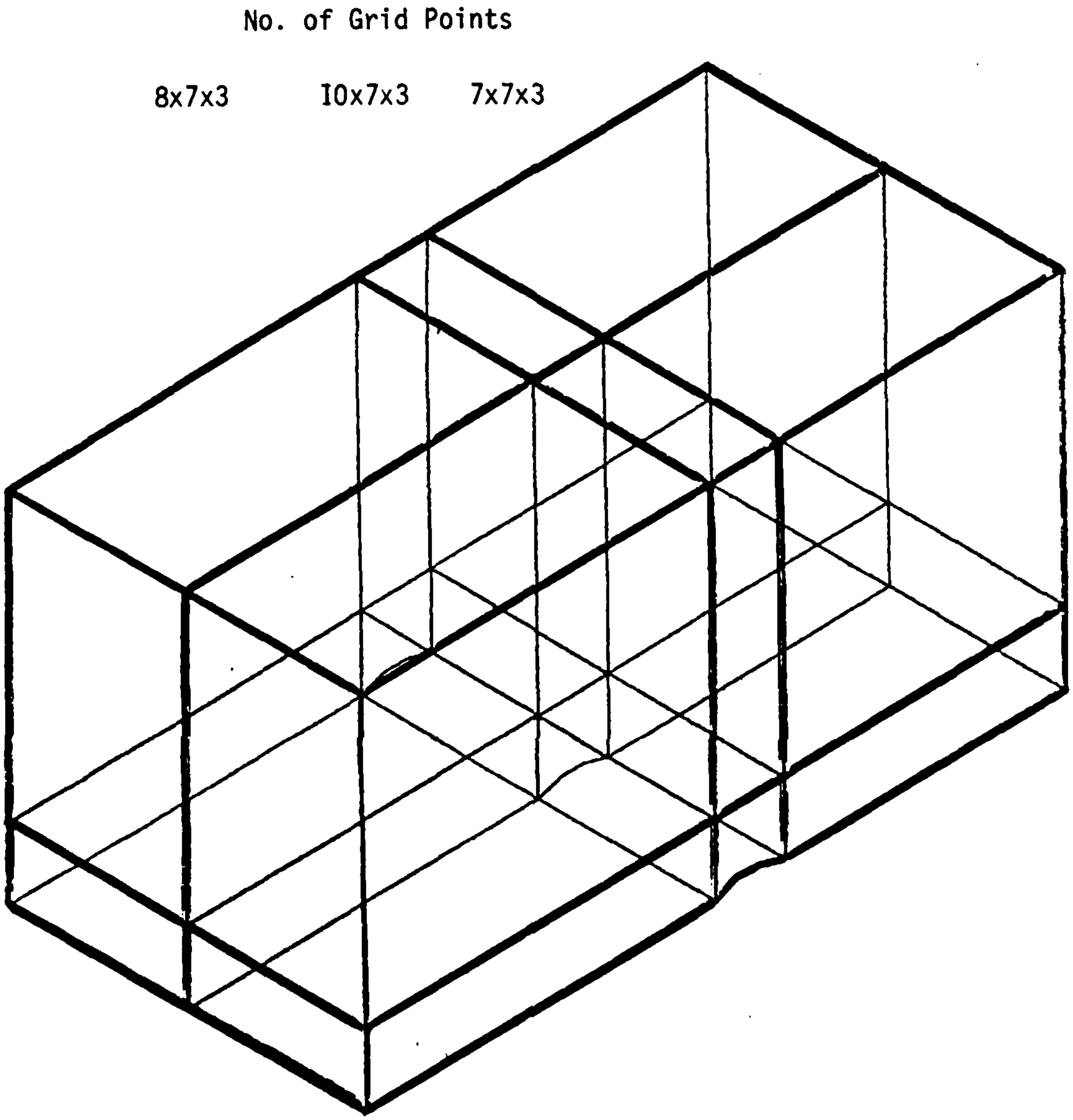


Figure 15. Twelve-block grid structure for the analysis of flow around a NACA0012 profile.

twelve intermediate surfaces. In this case, blocks of different sizes were placed in all three principal directions to test the robustness of the numerical scheme. The assembled grids for both types of block structures are identical in both cases and are shown in Figure 16. Both grids were considerably coarse with only ten elements over the airfoil but they still provided adequate accuracy for the solution of this simple program.

The numerical results obtained for an inlet profile of $M_{in} = 0.001$ from the three-block and twelve-block grid structures were compared with the exact solution of this problem, as shown in Figure 17. In both cases no relaxation was employed and both required about 35 iteration steps for convergence.

B. Analysis of Compressible Subsonic Flow Around a NACA0012 Airfoil ($M_{in} = 0.5$)

For the solution of this test problem the same three-block and twelve-block grid structures described in case A were employed. The obtained numerical results are compared in Figure 18 with results from a single-block C-type grid with (61) x (50) grid points.

The agreement with the one-block solution is very good considering the coarseness of the grid. The number of iterations required for convergence was about 40 for both cases and no relaxation factor was employed.

C. Analysis of Transonic Flow Around a NACA0012 Airfoil ($M_{in} = 0.83$)

Again two types of block structures were employed in the analysis. First, a three-block structure shown in Figure 14a was tested with blocks distributed only in the flow direction. The second case, shown in Figure 19, was a twelve-block structure involving blocks distributed in two dimensions. This time two blocks were employed over the airfoil surface to provide a complicated test case for locating the shock. The grids pro-

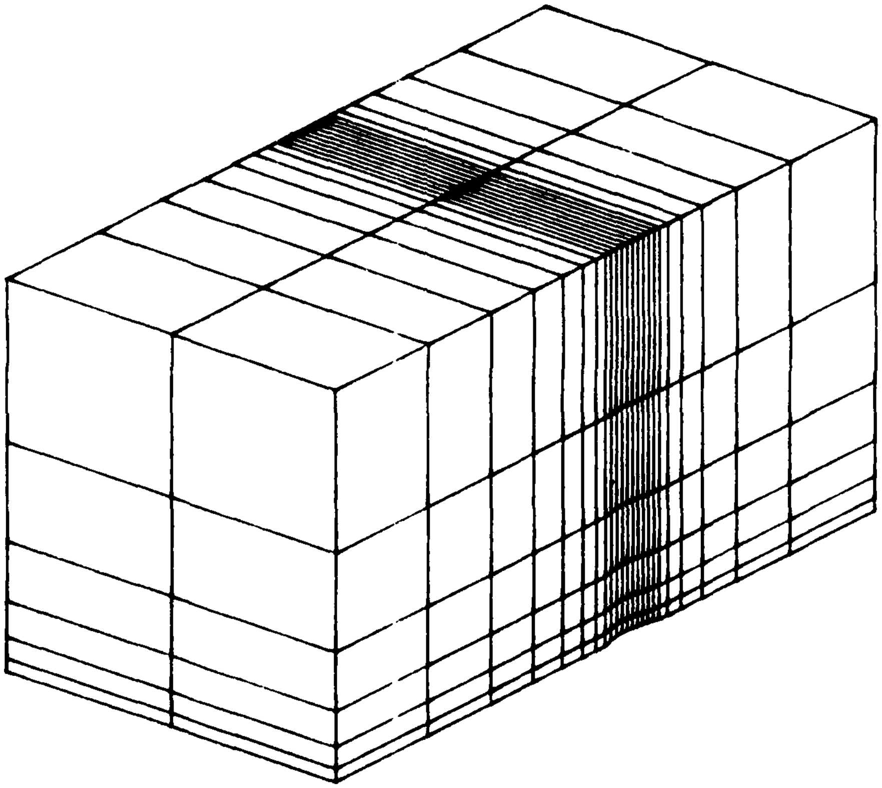


Figure 16. Assembled computational grids (12-block) for the analysis of flow around a NACA0012 profile.

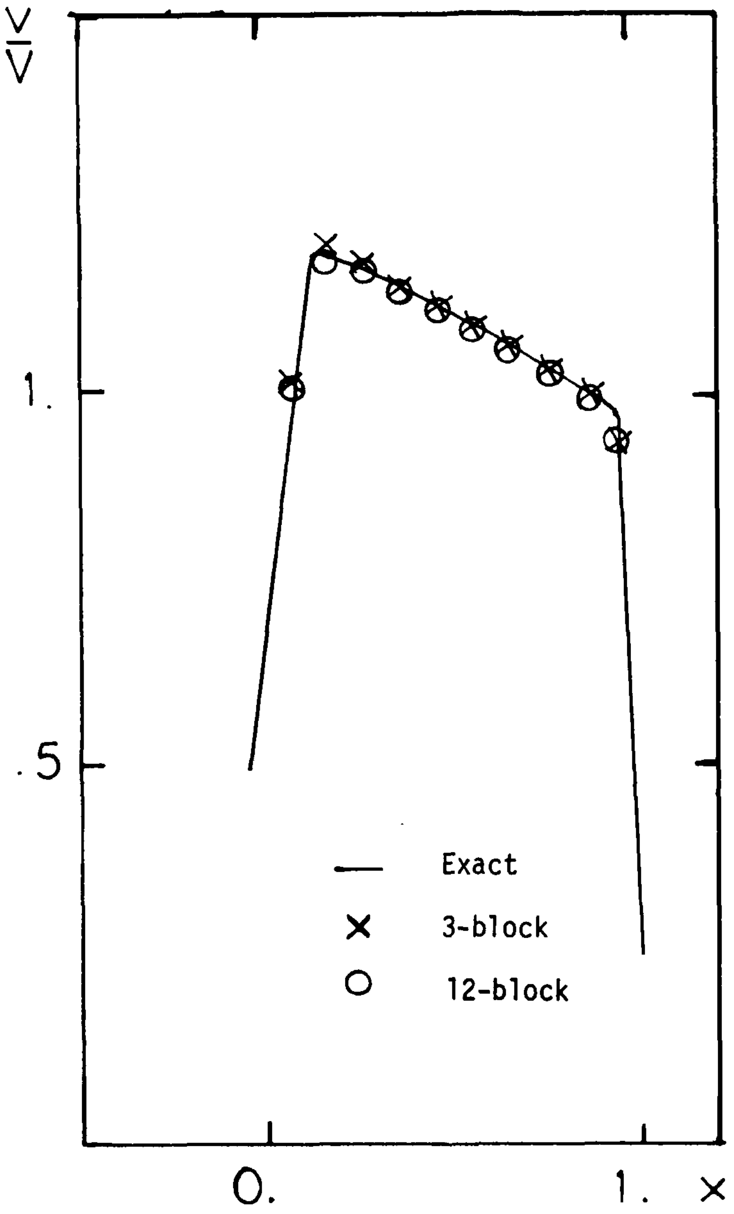


Figure 17. Incompressible flow (M_{in} = 0.001) results for flow around a NACA0012 profile.

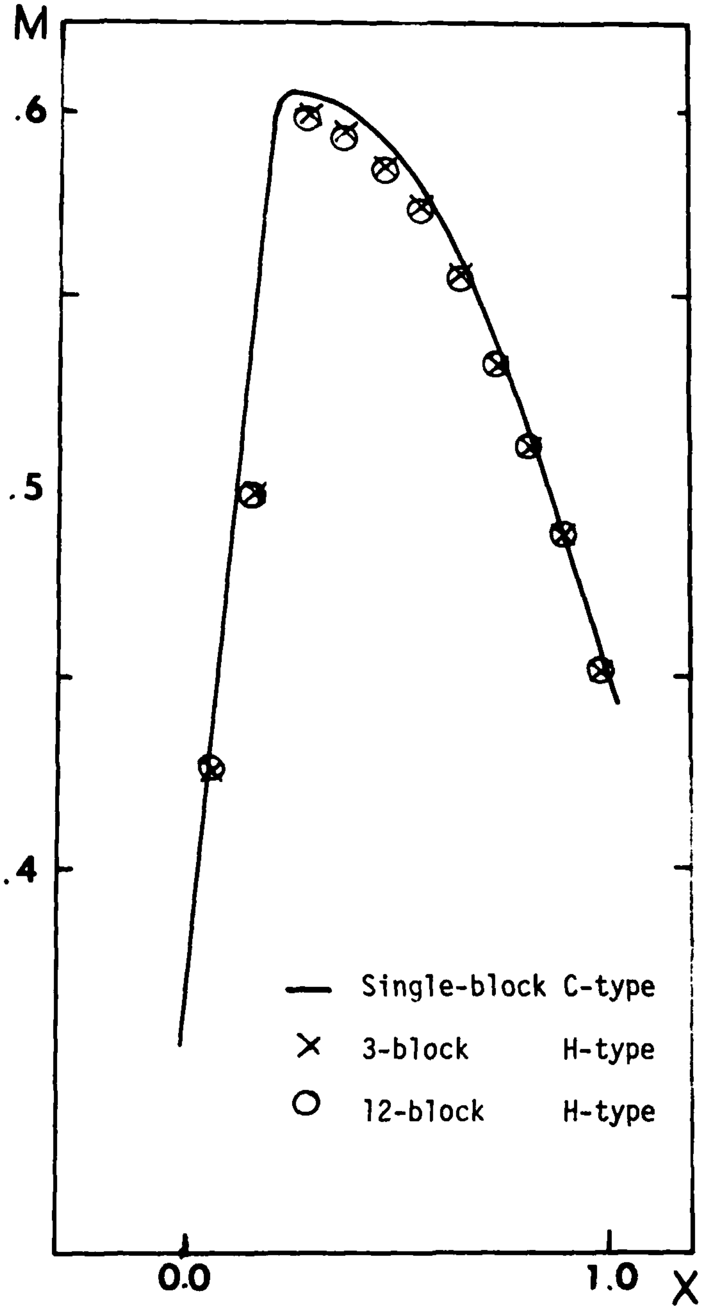


Figure 18. Compressible subsonic flow (M_{in} = 0.5) results for flow around a NACA0012 profile.

No. of Grid Points

9x11x3	13x11x3	16x11x3	16x11x3	13x11x3	9x11x3
9x 6x3	13x 6x3	16x 6x3	16x 6x3	13x 6x3	9x 6x3

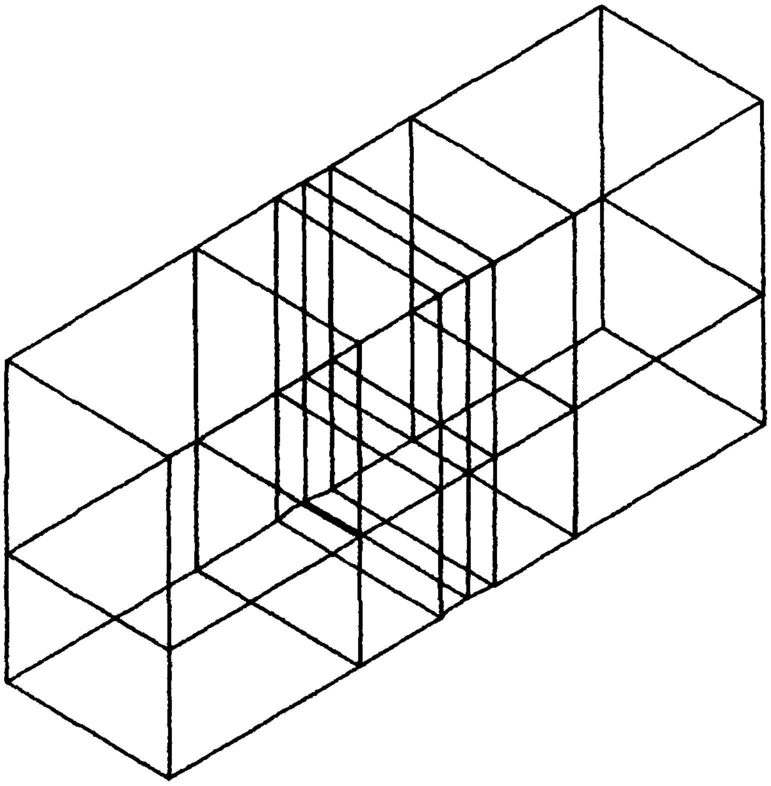


Figure 19. Twelve-block grid structure for the analysis of transonic flow (M_{in} = 0.83) around a NACA0012 profile.

duced for both block structures were identical but considerably finer than those employed in the ones used for incompressible flow problems. A cross-section of the grid is shown in Figure 20a. In Figure 20b the grid distribution over the airfoil surface can be seen. The grid employed (71)x(16x(3) = 3308 total grid points with a 30 element distribution over the airfoil. The distribution of elements for each block is shown in Figures 14a and 19.

The numerical results obtained from the three-block and twelve-block grid structures were compared with results obtained with a single-block C-type grid and are shown in Figure 21. Both results compare favorably with the single-block solution, obtained from the C-type grid shown in Figure 22. The number of iterations required for convergence was about 450 in both cases and involved a variable relaxation factor ω in the range of .01-.55.

D. Transonic Flow Around a Wing-body Configuration ($M_\infty = 0.90$, $\alpha = 3^0$)

A 24-block grid structure was chosen for analyzing the transonic flow problem around a wing-body configuration as shown in Figure 23. The details of the generated grid are shown in Figure 24. The number of grid points in each block is listed below:

Block No.	No. of Grid Points
B111	7x8x4
B121	14x8x4
B131	7x8x4
B112	7x8x4
B122	14x8x4
B132	7x8x4
B211	7x8x1
B221	14x8x1
B231	7x8x1
B212	7x8x1
B222	14x8x1
B232	7x8x1

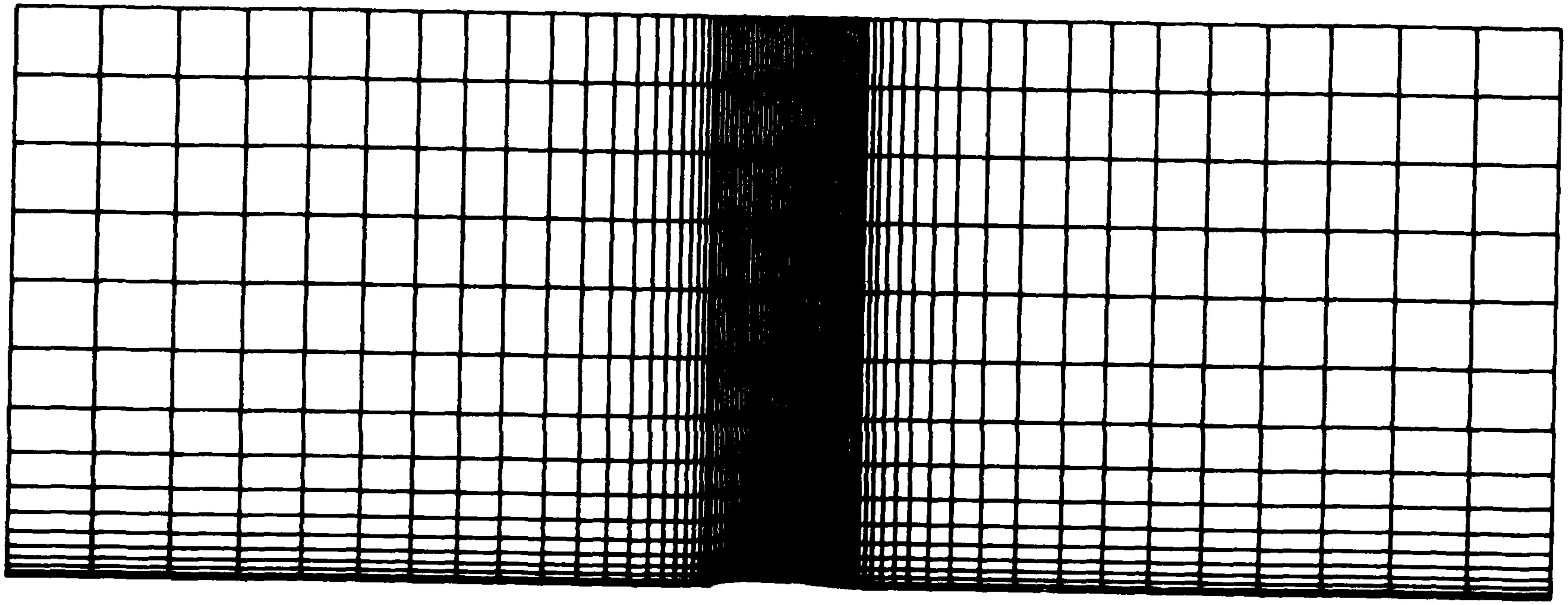


Figure 20a. A section of the grid for the analysis of transonic flow around a NACA0012 profile.

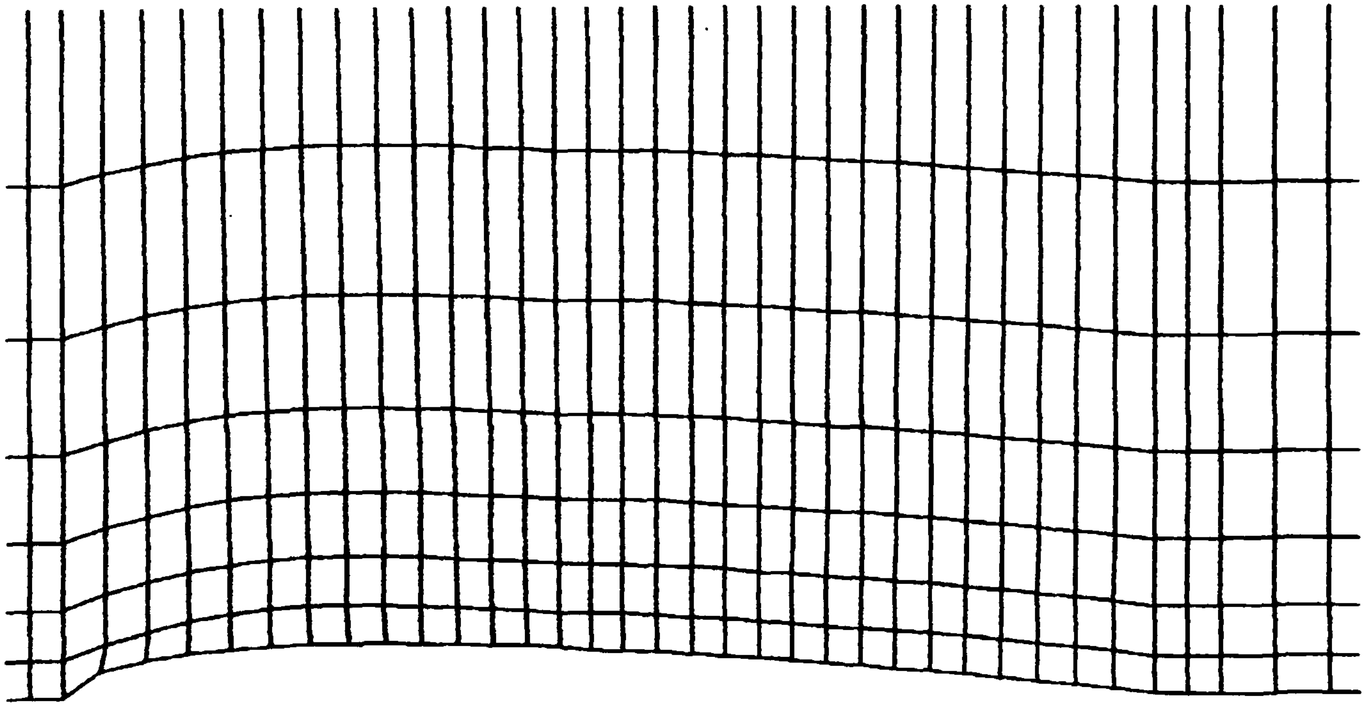


Figure 20b. Detail of the grid for the analysis of transonic flow around a NACA0012 profile.

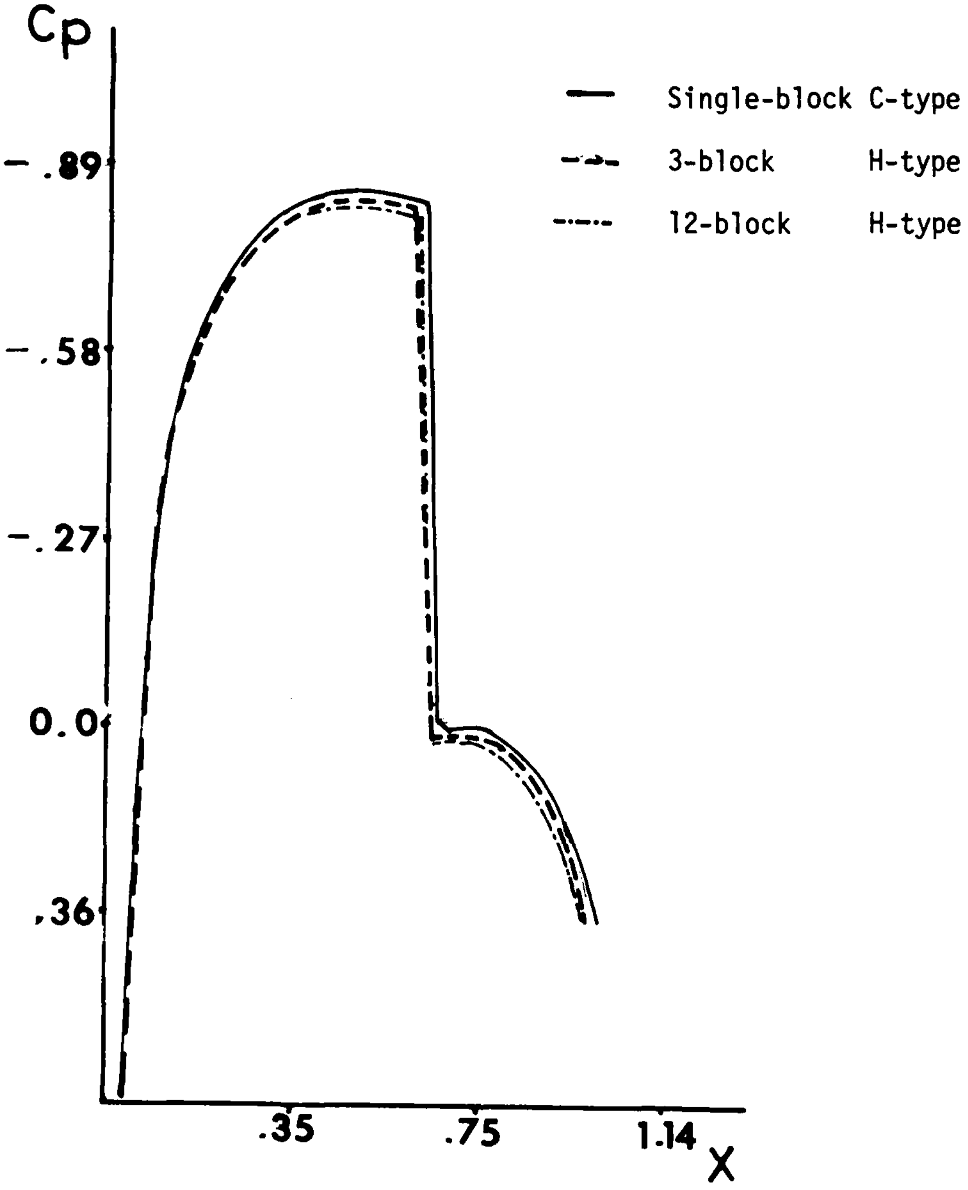


Figure 21. Transonic flow (M_{in} = 0.83) results for flow around a NACA0012 profile.

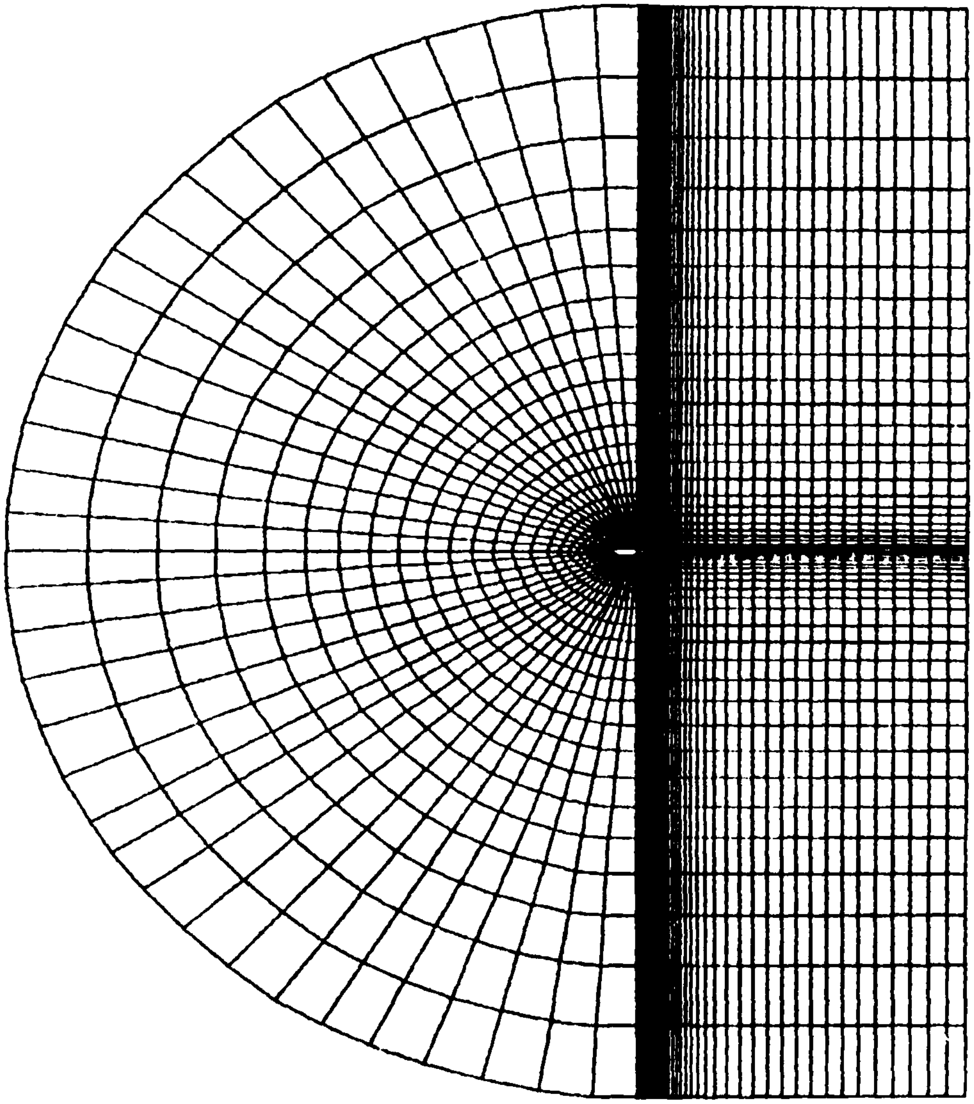


Figure 22. A section of the C-type grid for the analysis of transonic flow around a NACA0012 profile.

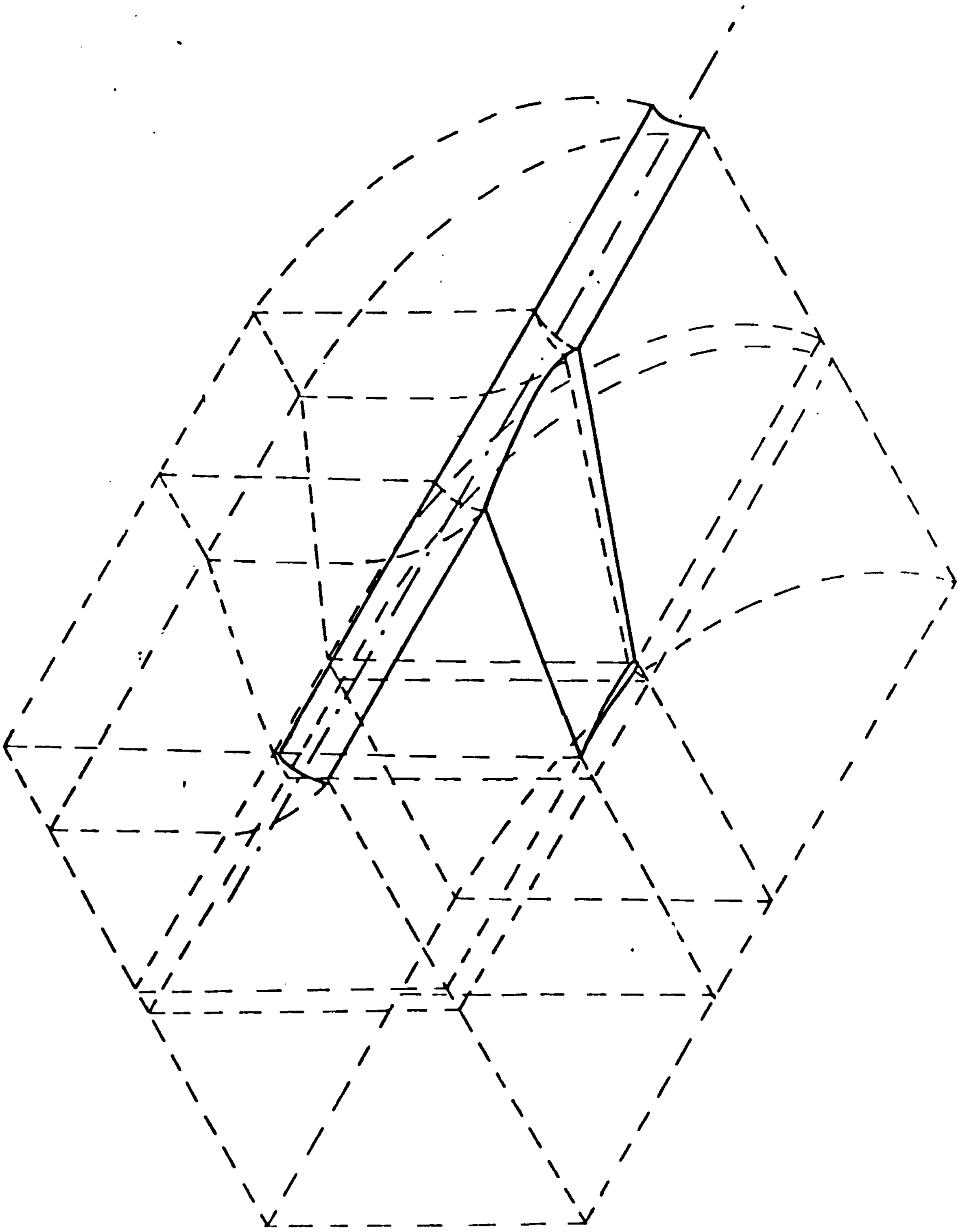


Figure 23a. Block structure for analyzing the transonic flow around the wing-body configuration.

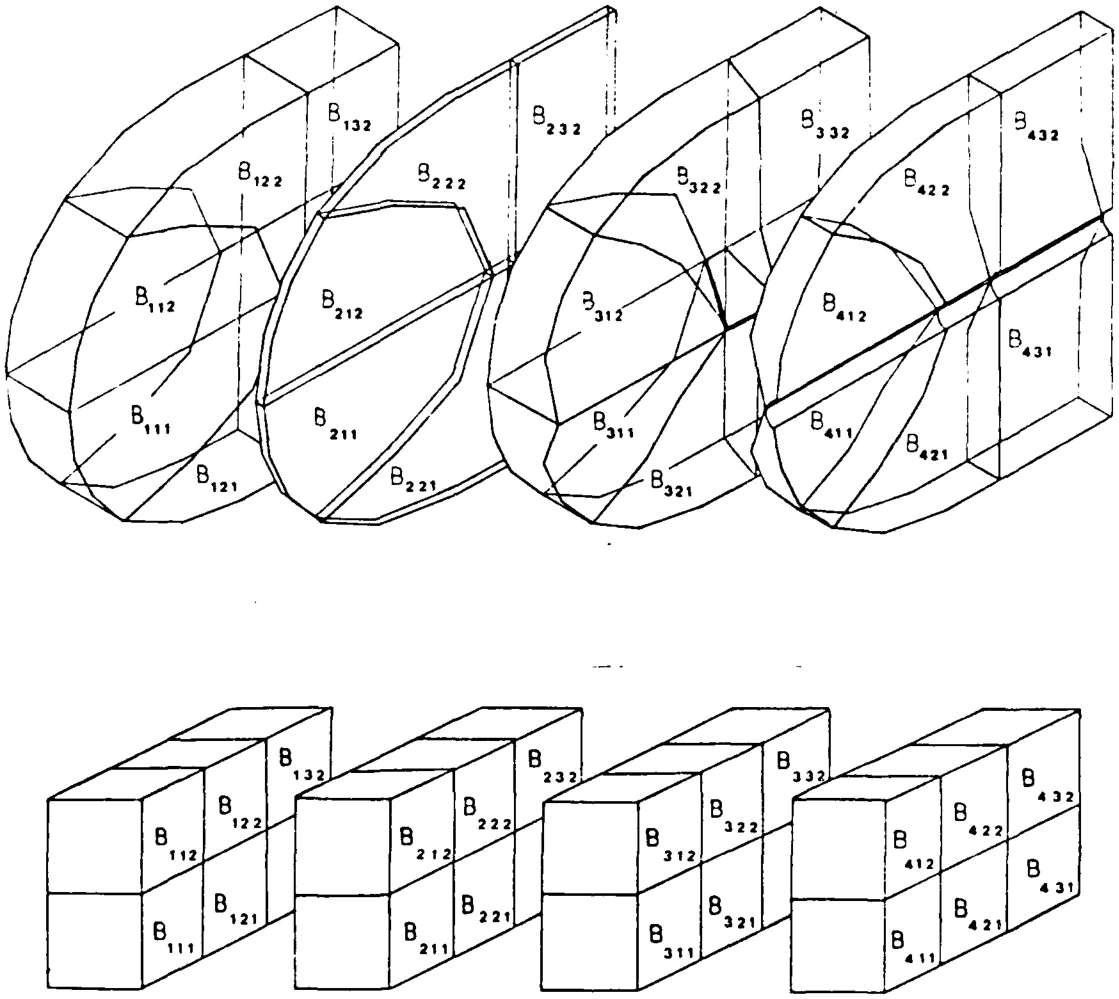


Figure 23b. Block representation of the flow domain in the real and computational spaces.

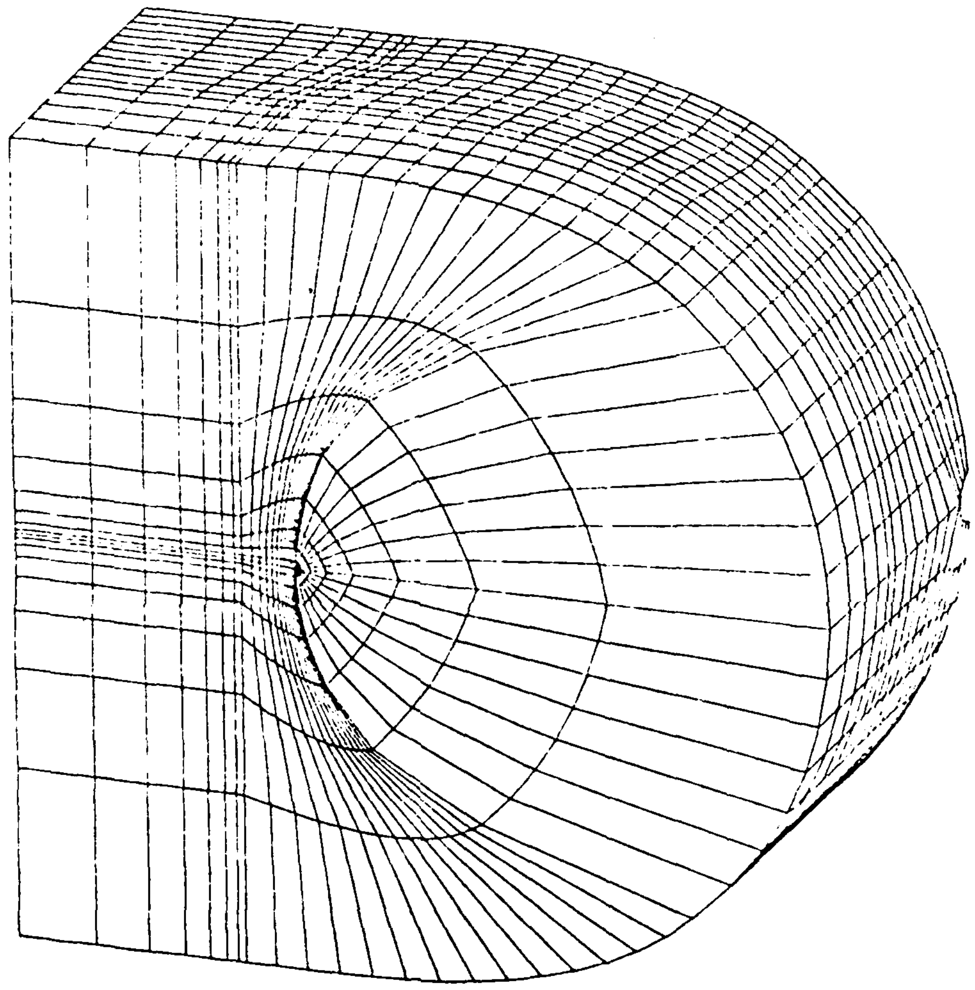


Figure 24a. Three-dimensional grid for the wing-body configuration.

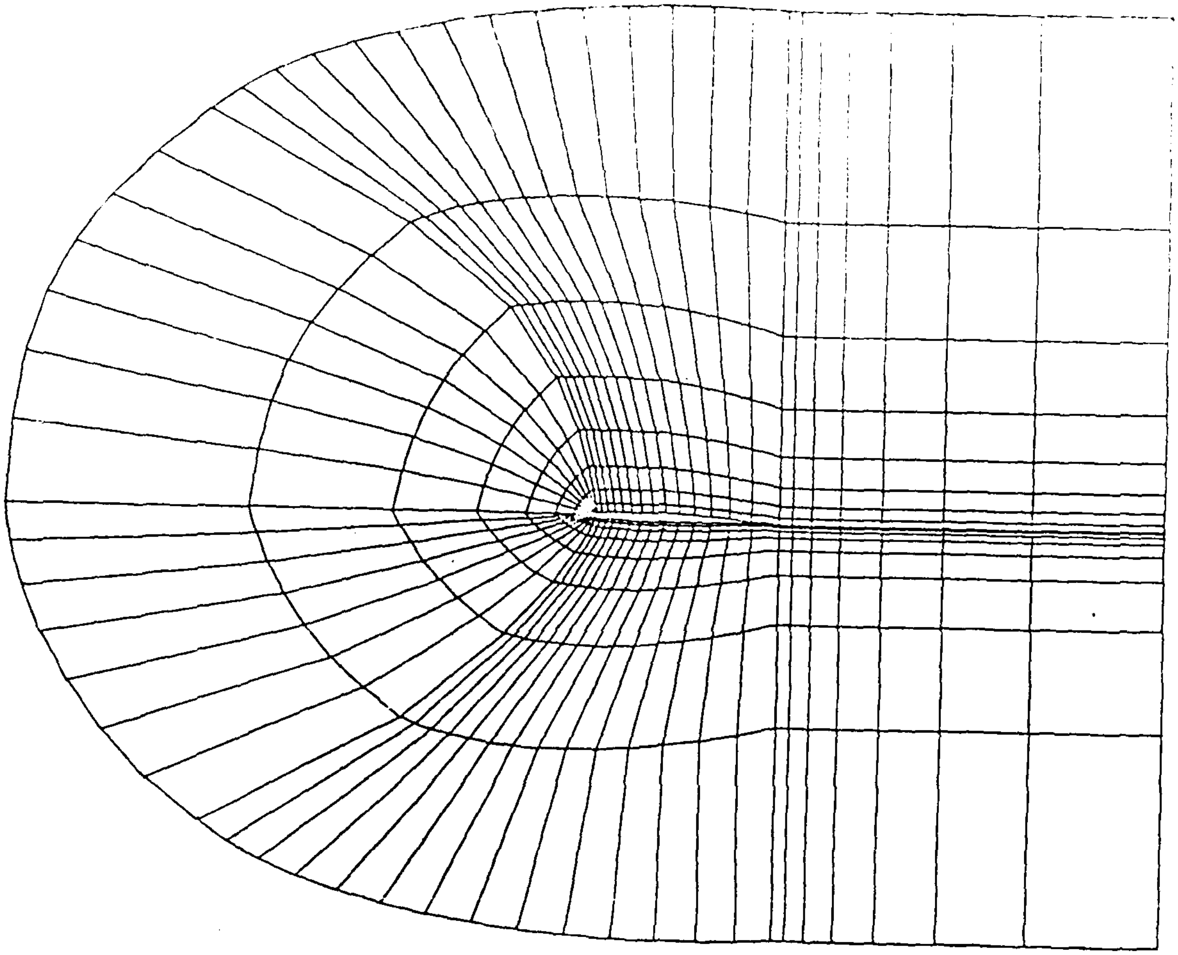


Figure 24b. Grid geometry along the wing cross-section.

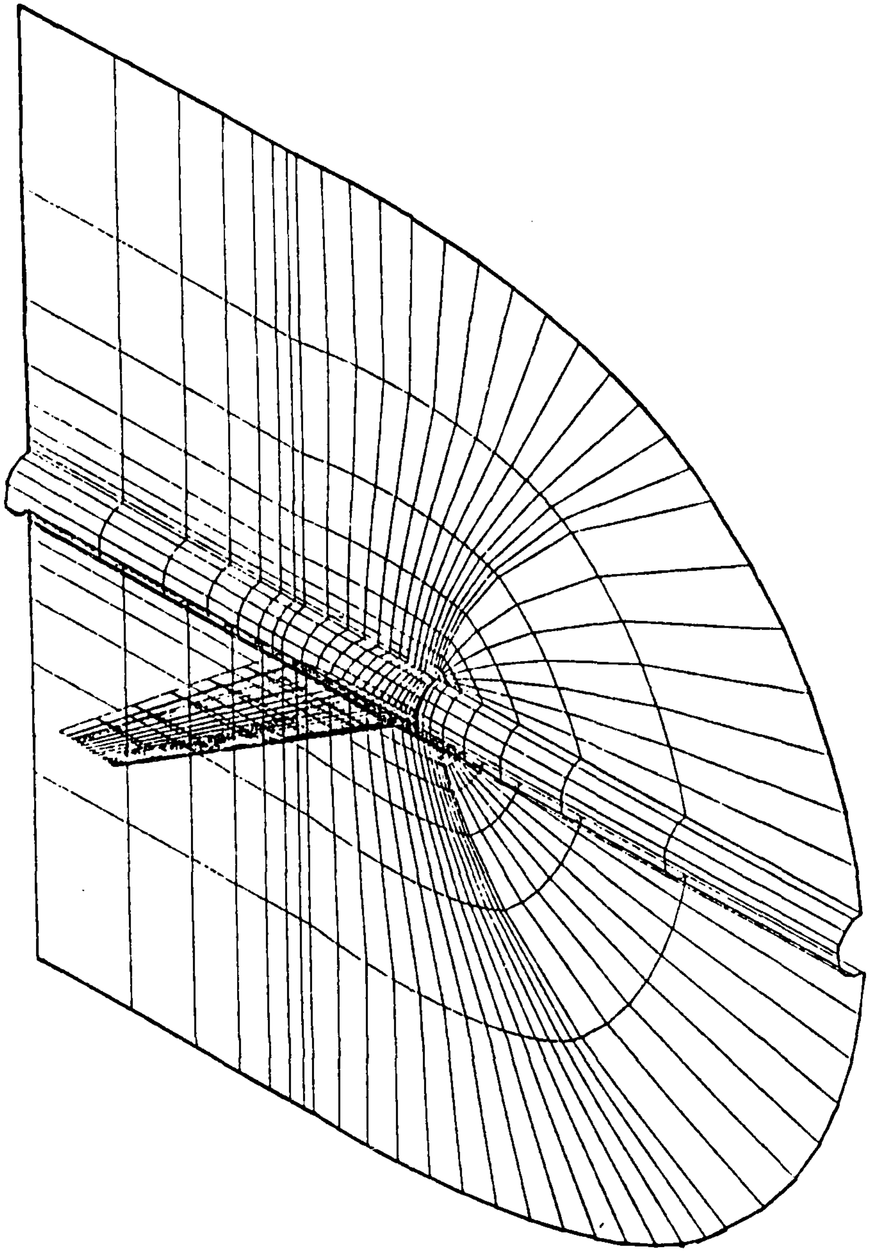


Figure 24c. Grid geometry around the wing-body surface and the outer boundary.

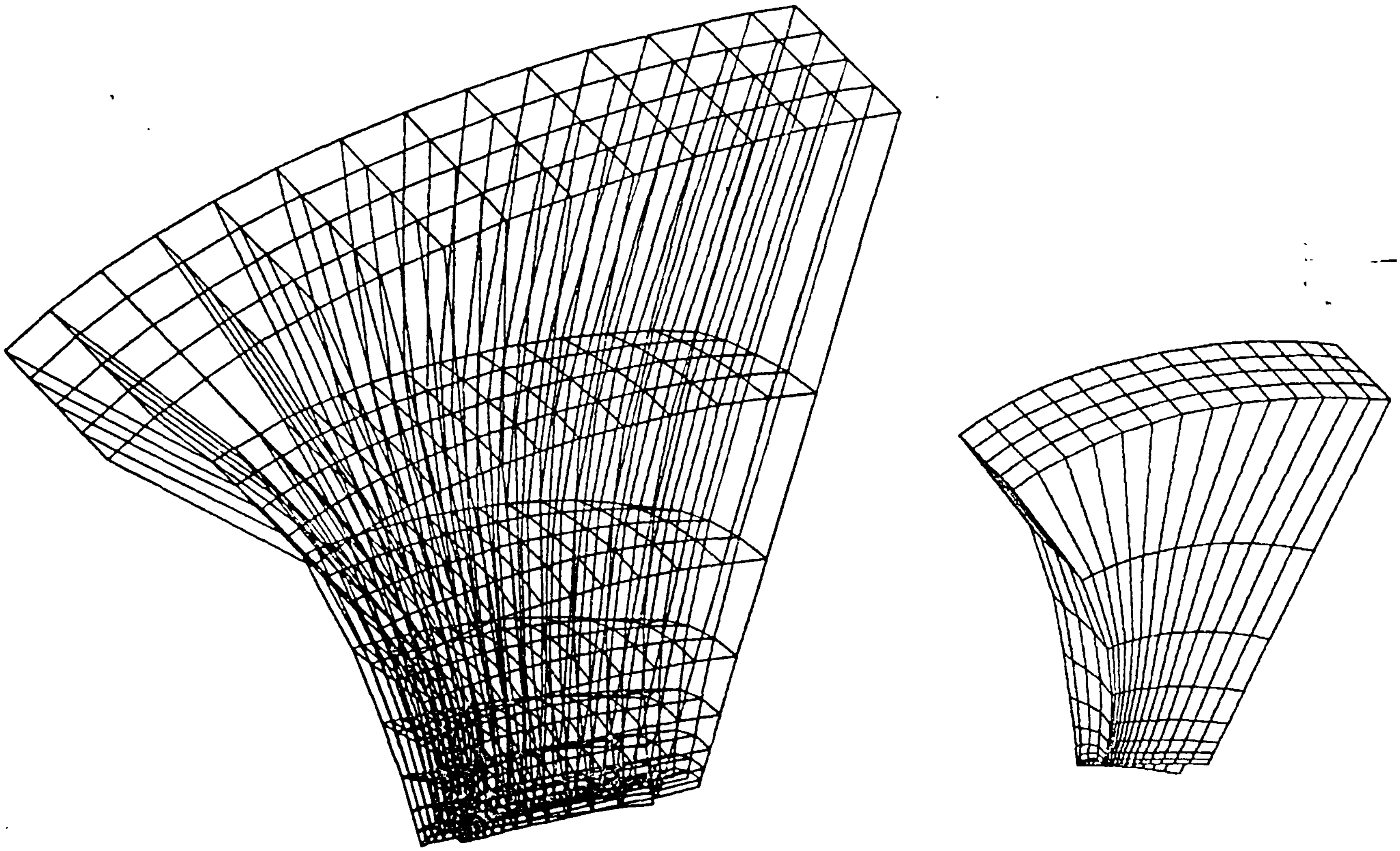


Figure 24d. Grid distribution for a single block with and without hidden lines (Block #B422 from Figure 23b).

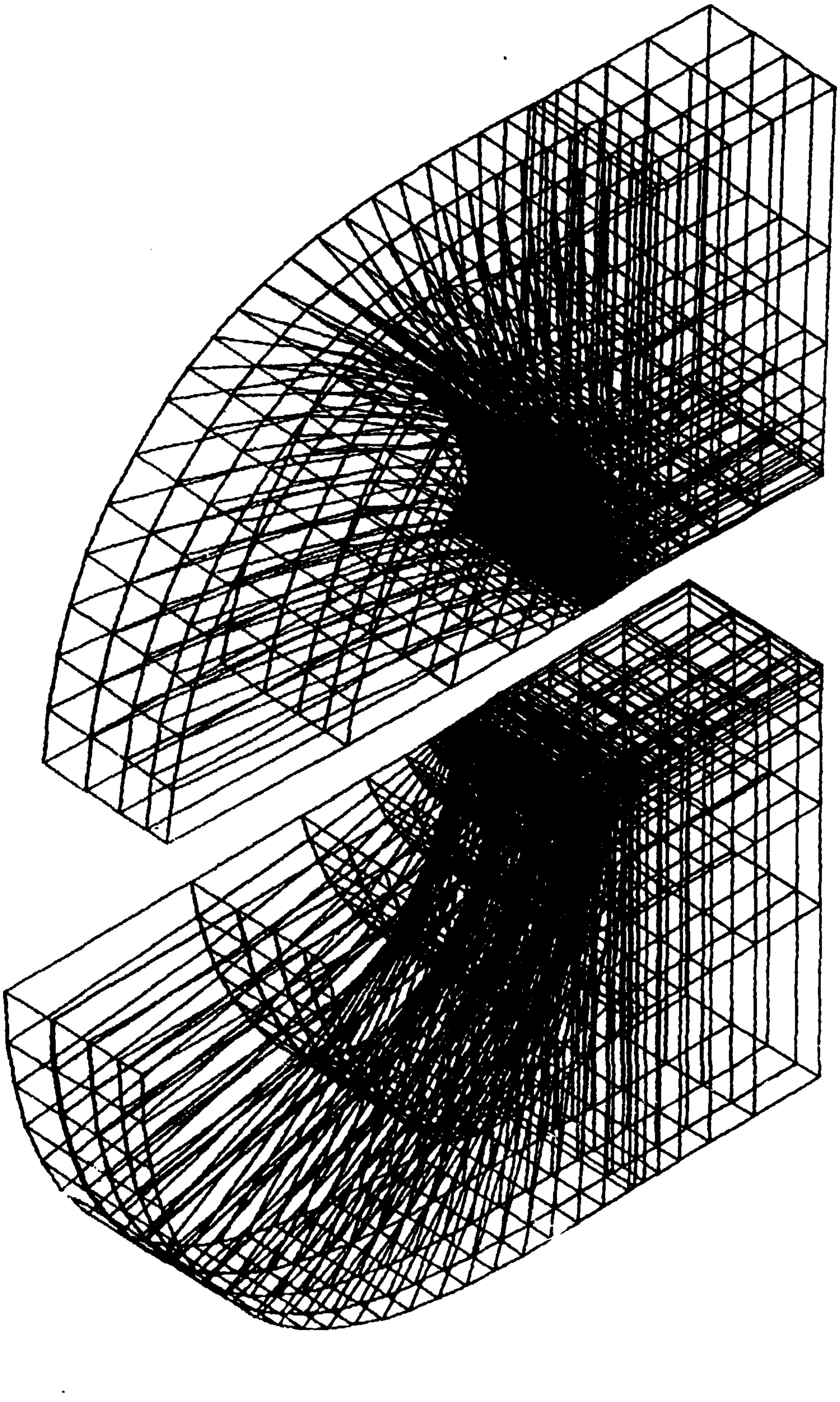


Figure 24e. Expanded view of the block grids for the first column of blocks for the wing-body problem.

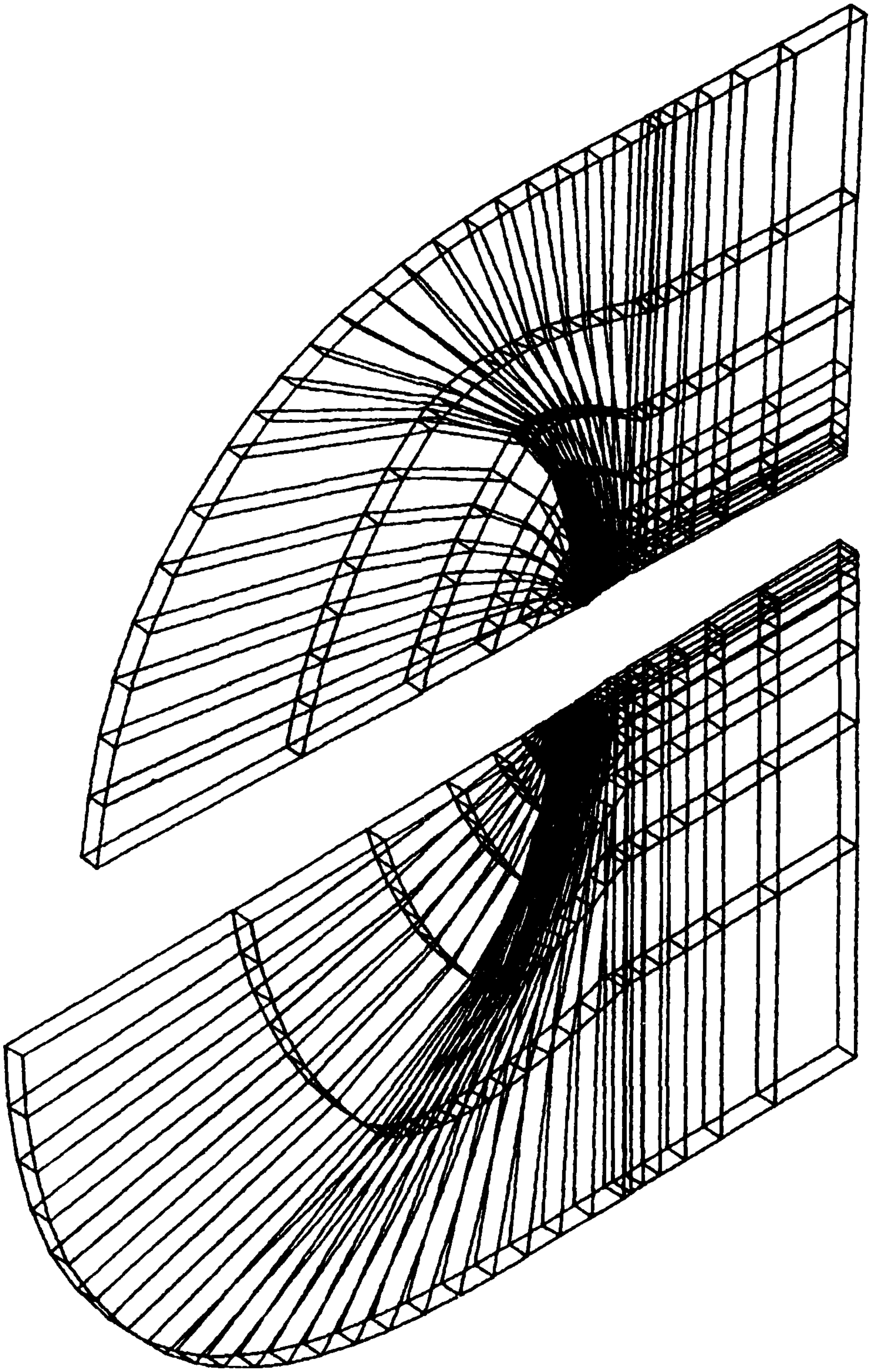


Figure 24f. Expanded view of the block grids for the second column of blocks for the wing-body problem.

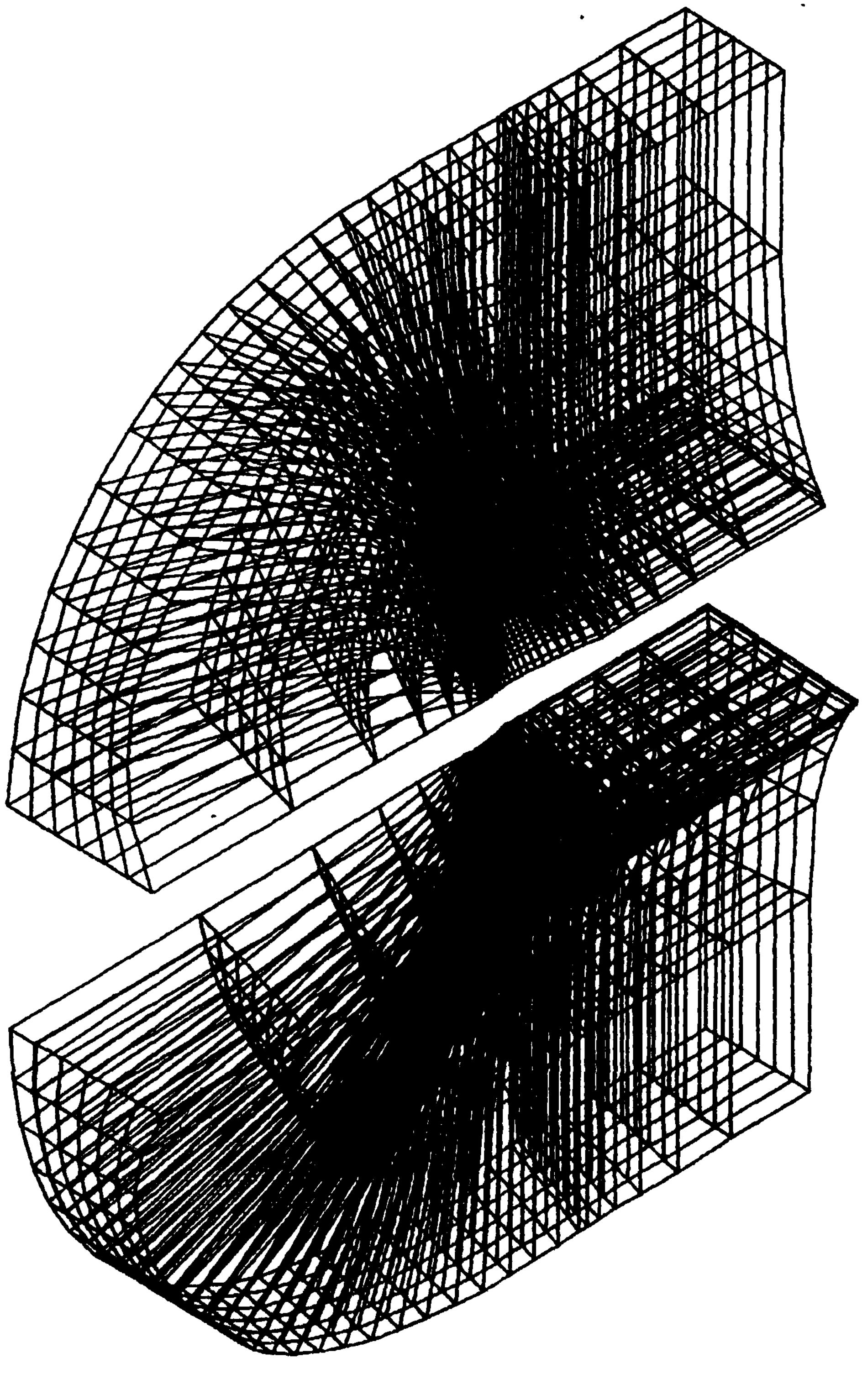


Figure 24g. Expanded view of the block grids for the third column of blocks for the wing-body problem.

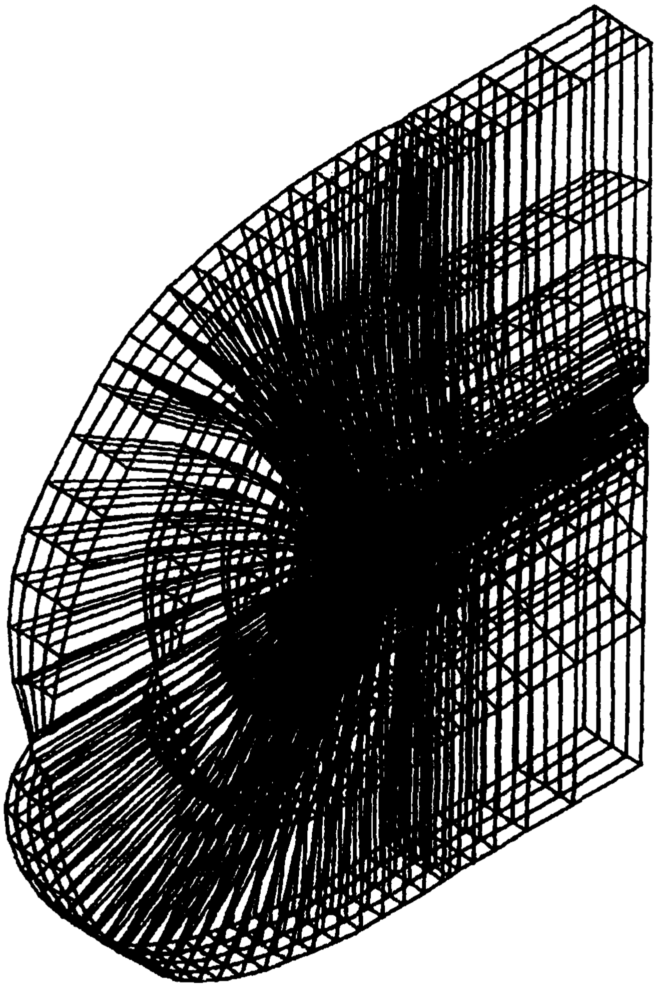


Figure 24h. Expanded view of the block grids for the fourth column of blocks for the wing-body problem.

Block No.	No. of Grid Points
B311	7x8x6
B321	14x8x6
B331	7x8x6
B312	7x8x6
B322	14x8x6
B332	7x8x6
B411	7x8x4
B421	14x8x4
B431	7x8x4
B412	7x8x4
B422	14x8x4
B432	7x8x4

The total number of grid points is 8207. Comparison of obtained results as obtained from the previous solution of the same grid as a single block [6] is shown in Figure 25.

As can be seen from these results, the 24-block solution reproduces the single block solution reasonably well. The differences between the two results can be attributed to the coarseness of the grid in terms of computing mass fluxes between the blocks.

The number of iterations required for convergence in this case was 550 and involved a variable relaxation factor ω in the range of .01-.75.

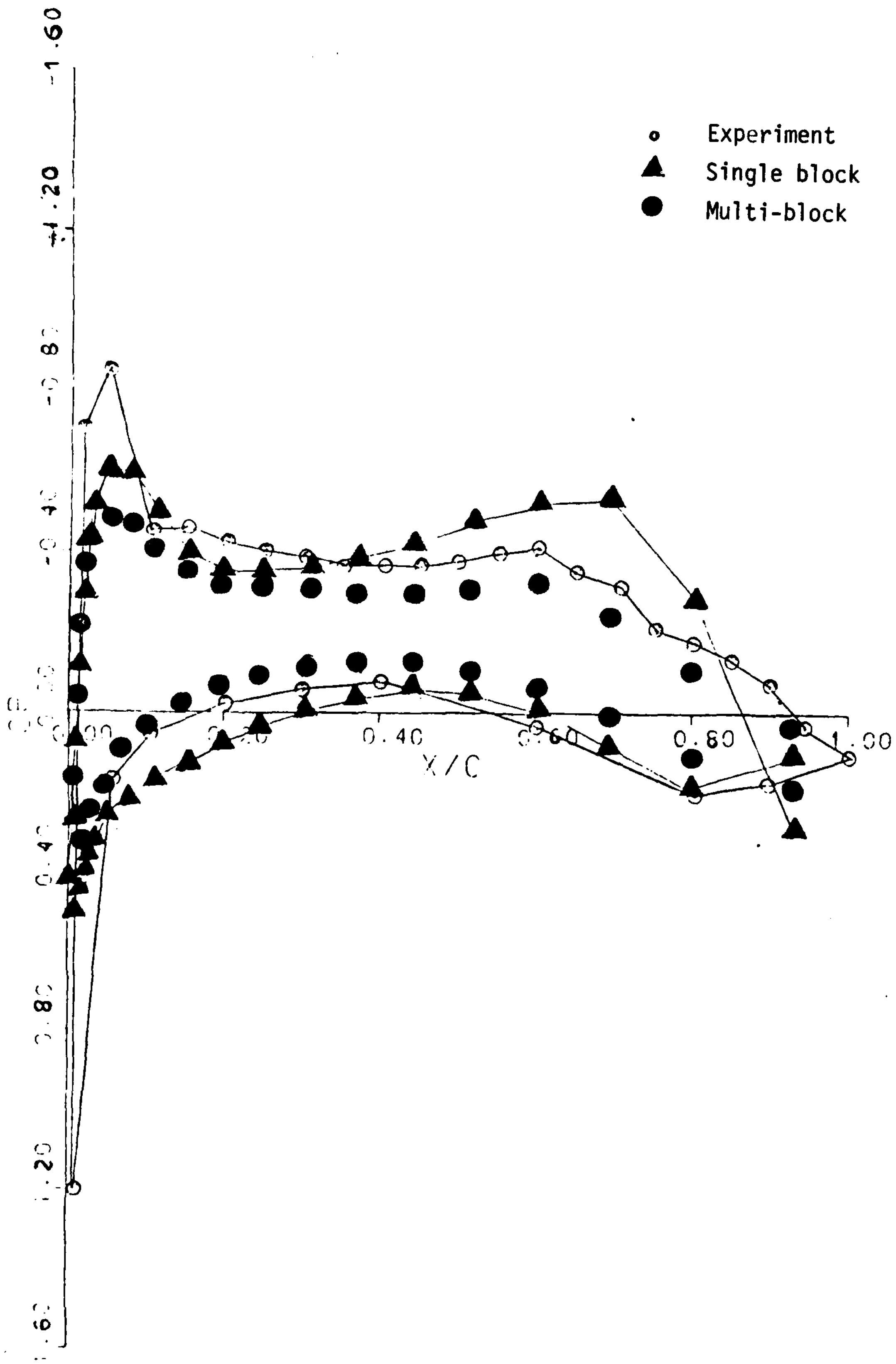


Figure 25. Comparison of the pressure distribution at the wing root for single block, multi-block and experimental results.

V. CONCLUSIONS

The procedure described in this report involves the solution of transonic potential flow equations using a block-structured approach. The basic objective of this research was to develop methodology for analyzing large flow problems involving complex geometries. The developed capabilities include,

- the generation of a computational grid around a complex geometry based on a block-structured grid generation scheme,
- the analysis of such a system by using a block-structured solution scheme for transonic potential flows.

The grid generation procedure was presented in previous publications. The block-structured solution scheme is presented in this report only for potential flows. It involves the solution of conservation of mass equation on a block-structured basis in terms of the velocity potential ϕ. Each block is stored independently and analyzed in the computer. Iterations involve solution of conservation of mass equation for each block as well as balancing the mass flux between the neighboring blocks.

The procedure is expandable to the solutions of Euler and Navier-Stokes equations. In the case of Euler equations, equation for the conservation of entropy will also have to be solved. This again will involve the solution of the equation for each block as well as balancing the entropy to the neighboring blocks in the flow direction. Such an approach provides the capability of solving different types of equations approximating the flow field on the same grid but for different blocks. For a complex aircraft configuration, the capability to solve potential, Euler or Navier-Stokes equations at different flow regions is necessary considering the present computer hardware capabilities and also the complexity of the problem.

As stated above, the potential flow solution of transonic flows requires the only the treatment of the conservation of mass equation. In terms of solving Euler or Navier-Stokes equations, the same equation has to be treated. In fact, the elliptic nature of the conservation of mass equation governs the convergence rate of relaxation schemes for all three types of formulations. Therefore, the efficiency of the computational scheme in terms of solving the conservation of mass equation for large systems is of general importance for solving Euler and Navier-Stokes equations.

Currently the method is being extended to the solution of Euler equations including solving coupled potential and Euler solutions at different flow regions. Another important consideration is the application of the numerical procedure in a parallel processing environment. As described in this report, the processing of blocks and surfaces can be performed in any order. This provides considerable flexibility in defining an optimum solution procedure for a given parallel processor hardware configuration.

VI. REFERENCES

1. Rakich, J. V., Introduction to the Proceedings of the Symposium on Computational Fluid Dynamics, AIAA Computational Fluid Dynamics Conference, Palm Springs, California, July 19-20, 1973.
2. Proceedings of the AIAA Computational Fluid Dynamics Conference, Danvers, Massachusetts, July 13-20, 1983.
3. Chima, R. V. and Johnson, G. M., Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid algorithm, Proceedings of the AIAA Computational Fluid Dynamics Conference, Danvers, Massachusetts, July 13-20, 1985.
4. Ruppert, P. E. and Lee, K. D., Patched Coordinate Systems, Numerical Grid Generations, Ed. by J. F. Thompson, North Holland, 1982.
5. Ecer, A., Spyropoulos, J. and Maul, J., A Block-Structured Finite Element Grid Generation Scheme for the Analysis of Three-Dimensional Transonic Flows, AIAA Paper No. 84-0004, AIAA Journal, September 1985.
6. Ecer, A., Citipitioglu, E. and Bhutta, B., Design of Finite Element Grids for the Computation of Three-Dimensional Transonic Flow Around a Wing, AIAA Paper No. 82-1019.
7. Ecer, A. and Akay, H. U., A Finite Element Formulation of Euler Equations for the Solution of Steady Transonic Flows, AIAA Journal, Vol. 21, No. 3, March 1983.
8. Coleman, R. M., Conservation of Boundary Fitted Coordinate Systems Using Sepmented Computational Regions, Numerical Grid Generations, Ed. by J. F. Thompson, North Holland, 1982.

9. Akay, H. U. and Ecer, A., Finite Element Formulation of Rotational Transonic Flow Problems, Finite Elements in Fluids, Ed. by R. H. Gallagher, et al., John Wiley, 1984, pp. 173-195.

10. Akay, H. U., Ecer, A. and Willhite, P., Finite Element Solutions of Euler Equations for Lifting Airfoils, AIAA Paper No. 85-0294, to be published in the AIAA Journal.